Corresponding to D3

FILE CIPHERING SYSTEM AND ITS CONTROL METHOD, AND CIPHER FILE RECEPTION SYSTEM AND ITS CONTROL METHOD

Publication number: JP9251426 (A)

Publication date: 1997-09-22

Inventor(s): ARAI MASATO; ITO HIROMICHI; SUZAKI SEIICHI; UMEKI HISASHI; OTSU YUTAKA; MORIFUJI HAJIME; SHIMIZU MAYUKO

Applicant(s): HITACHI LTD

Classification:

- international: ***G06F12/14; G06F12/00; G06F13/00; G06F21/24;*** **G06F12/14; G06F12/00; G06F13/00; G06F21/00;** (IPC1-7): G06F12/14; G06F12/00; G06F13/00

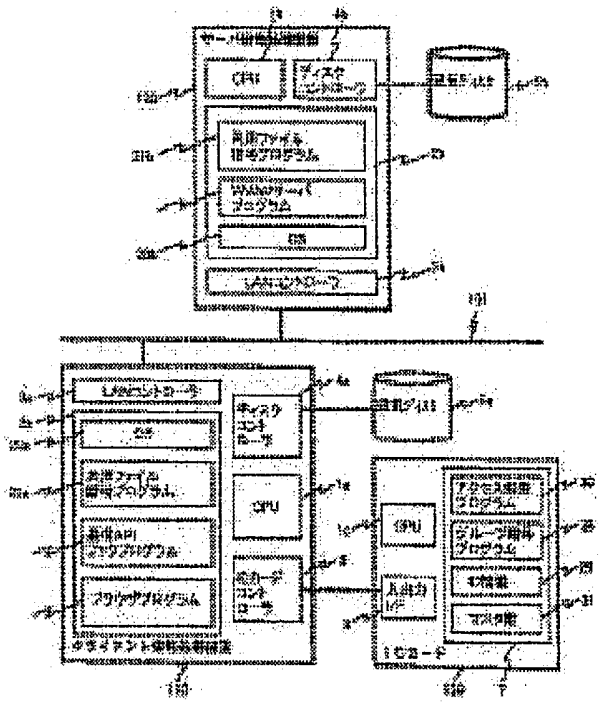
- European:

Application number: JP19960182977 19960712

Priority number(s): JP19960182977 19960712; JP19960002046 19960110

Abstract of **JP 9251426 (A)**

PROBLEM TO BE SOLVED: To provide a safe cipher file reception system and its method which need only one cipher file even when plural users access the same WWW(world wide web). SOLUTION: This cipher file reception system includes a server information processor 120 which includes a shared file cipher program 21b and previously ciphers the information transmitted through a WWW to store them in a magnetic disk 5b, and a client information processor 110 which includes a shared file cipher program 21a and a communication API hook program. Then the system hooks the communication API call sent from a browser program and automatically decodes the received file data.



D3

(19) 日本国特許庁（ＪＰ）　(12) 公開特許公報（Ａ）　(11) 特許出願公開番号

特開平9－251426

(43) 公開日　平成９年(1997) ９月22日

(51) Int. Cl. [6]	識別記号	ＦＩ		
G06F 12/14	320	G06F 12/14	320	B
12/00	537	12/00	537	H
	545		545	F
13/00	353	13/00	353	M

審査請求　未請求　請求項の数39　ＯＬ　（全33頁）

(21) 出願番号　特願平8－182977

(22) 出願日　平成８年(1996) ７月12日

(31) 優先権主張番号　特願平8－2046

(32) 優先日　平８(1996) １月10日

(33) 優先権主張国　日本（ＪＰ）

(71) 出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72) 発明者　荒井　正人
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内

(72) 発明者　伊藤　浩道
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内

(72) 発明者　洲崎　誠一
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内

(74) 代理人　弁理士　沼形　義彰　（外１名）

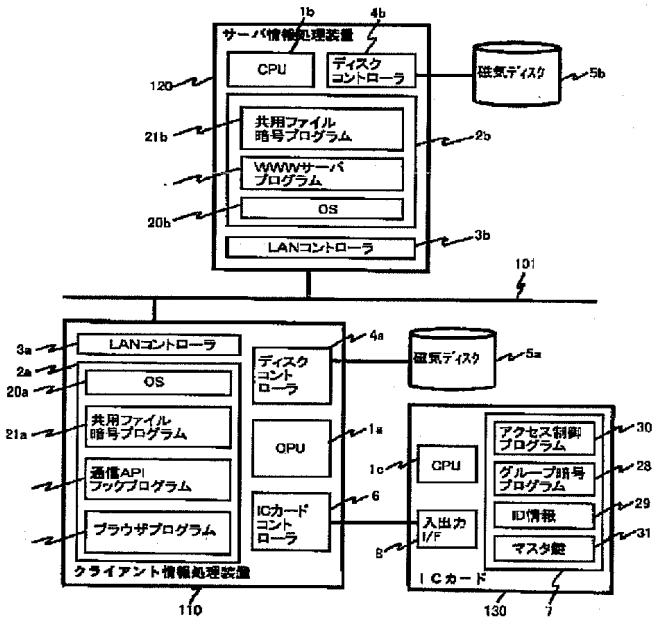
最終頁に続く

(54) 【発明の名称】ファイル暗号化システム及びその制御方法ならびに暗号ファイル受信システム及びその制御方法

(57) 【要約】

【課題】　複数のユーザーが同一のファイルをWWWでアクセスする場合にも、暗号ファイルが一個で済み、且つ安全な暗号ファイル受信システム及びその制御方法を提供する。

【解決手段】　共用ファイル暗号プログラム２１ｂを有し、WWWで発信する情報を予め暗号化して磁気ディスク５ｂに格納したサーバ情報処理装置１２０と、共用ファイル暗号プログラム２１ａと通信ＡＰＩフックプログラム２７を設けたクライアント情報処理装置１１０とからなり、ブラウザプログラム１９からの通信ＡＰＩ呼び出しをフックし、受信したファイルデータを自動的に復号化するようにした暗号ファイル受信システム。



1

【特許請求の範囲】

【請求項１】　少なくともファイルへの入出力制御機能を持つオペレーティングシステム（ＯＳ）と、前記ファイルを格納する記憶装置あるいはネットワークで接続された第二の情報処理装置上の記憶装置へのファイル格納手段とを具備する情報処理装置からなるファイル暗号化システムにおいて、暗号化を行う対象ディレクトリを予め設定しファイルに記憶する暗号化ディレクトリ記憶手段と、前記ＯＳ上で動作するアプリケーションプログラムから前記ディレクトリ内のファイルへの書き込みデータを暗号化する暗号化手段とを設けたことを特徴とするファイル暗号化システム。

【請求項２】　前記暗号化ディレクトリ記憶手段は、前記情報処理装置を利用するユーザー毎に、独立した複数の暗号化対象ディレクトリを設定可能であることを特徴とする請求項１記載のファイル暗号化システム。

【請求項３】　前記暗号化ディレクトリ記憶手段は、前記対象ディレクトリ毎に復号可能なユーザーのリストを記憶することを特徴とする請求項１または請求項２記載のファイル暗号化システム。

【請求項４】　前記ユーザーのリストを記憶したファイルを、前記対象ディレクトリ毎に前記対象ディレクトリ内に作成することを特徴とする請求項３記載のファイル暗号化システム。

【請求項５】　前記暗号化手段は、当該ディレクトリに暗号化対象ディレクトリであることを示すファイルがある場合にファイルへの書き込みデータを暗号化することを特徴とする請求項１記載のファイル暗号化システム。

【請求項６】　前記ユーザーのリストを暗号化し記憶することを特徴とする請求項３ないし請求項５のいずれか記載のファイル暗号化システム。

【請求項７】　前記アプリケーションプログラムから見た論理的ファイル名と、前記暗号化手段によって格納される物理的ファイル名とが異なることを特徴とする請求項１ないし請求項６のいずれか記載のファイル暗号化システム。

【請求項８】　少なくともファイルへの入出力制御機能を持つオペレーティングシステム（ＯＳ）と、前記ファイルを格納する記憶装置あるいはネットワークで接続された第二の情報処理装置上の記憶装置へのファイル格納手段とを具備する情報処理装置からなるファイル暗号化システムにおいて、暗号化を行う対象ディレクトリを予め設定しファイルに記憶する暗号化ディレクトリ記憶手段と、アプリケーションプログラムから前記ディレクトリ内のファイルへの書き込みデータの暗号化と読み出しデータの復号化とを行う暗号復号手段とを設けたことを特徴とするファイル暗号化システム。

【請求項９】　前記暗号復号手段は、アプリケーションプログラムからのファイル書き込み或いは読み出しのサイズとは独立したサイズの平文ブロックバッファと暗号

2

文ブロックバッファとを具備し、暗号化および復号化の処理は、前記ブロックバッファ上の全データを一度に処理することによって行うことを特徴とする請求項８記載のファイル暗号化システム。

【請求項１０】　前記暗号化手段あるいは復号化手段は、前記アプリケーションプログラムから前記ＯＳに発行されるファイル操作要求をフックするフック手段によって呼び出されることを特徴とする請求項１ないし請求項９のいずれか記載のファイル暗号化システム。

【請求項１１】　前記暗号化手段は、復号を許可するユーザーのリストの文字列をハッシュ関数によって圧縮することによって作成した数値を暗号化の鍵として用いることを特徴とする請求項１ないし請求項１０のいずれか記載のファイル暗号化システム。

【請求項１２】　前記暗号化手段は、復号を許可するユーザーのリストの文字列をハッシュ関数によって圧縮することによって作成した数値で暗号化の鍵を暗号化し、該暗号化鍵を暗号化ファイルに付加することを特徴とする請求項１なし請求項１０のいずれか記載のファイル暗号化システム。

【請求項１３】　少なくともファイルへの入出力制御機能を持つオペレーティングシステム（ＯＳ）と、前記ファイルを格納する記憶装置あるいはネットワークで接続された第二の情報処理装置上の記憶装置へのファイル格納手段と、前記ファイルを暗号化するファイル暗号化手段とを具備する情報処理装置からなるファイル暗号化システムの制御方法において、前記ファイル暗号化手段が、アプリケーションプログラムが解釈し処理する文法に従って記述された平文ファイルのデータ暗号化に際し、前記文法における前記アプリケーションプログラム処理時に処理対象外として無視する注釈文とするための記述子を前記暗号化後のデータに付加し格納することを特徴とするファイル暗号化システムの制御方法。

【請求項１４】　前記暗号化後のデータに前記注釈文の終了を示す記述子が存在しないような暗号化鍵を選択し使用することを特徴とする請求項１３記載のファイル暗号化システムの制御方法。

【請求項１５】　ファイルが暗号文のファイルであることを前記アプリケーションプログラムが表示するための記述文を、前記暗号化後のデータを含む前記注釈文の前あるいは後ろに付加し格納することを特徴とする請求項１３又は請求項１４記載のファイル暗号化システムの制御方法。

【請求項１６】　少なくともファイルへの入出力制御機能を持つオペレーティングシステム（ＯＳ）と、前記ファイルを格納する記憶装置あるいはネットワークで接続された第二の情報処理装置上の記憶装置へのファイル格納手段と、前記記憶装置に格納されたファイルを復号化するファイル復号化手段とを具備する情報処理装置からなるファイル暗号化システムの制御方法において、前記

ファイル復号化手段が、アプリケーションプログラムが解釈し処理する文法に従って記述された暗号ファイルのデータ復号化に際し、前記文法における前記アプリケーションプログラム処理時に処理対象外として無視する注釈文とするための記述子によって注釈文とされたデータ部分を復号化し、該復号化後のデータを暗号化前の平文ファイルのデータとして格納することを特徴とするファイル暗号化システムの制御方法。

【請求項１７】　前記暗号化手段あるいは復号化手段は、前記アプリケーションプログラムから前記ＯＳに発行されるファイル操作要求をフックするフック手段によって呼び出されることを特徴とする請求項１３ないし請求項１６のいずれか記載のファイル暗号化システムの制御方法。

【請求項１８】　前記暗号化手段は、復号を許可するユーザーのリストの文字列をハッシュ関数によって圧縮することによって作成した数値を暗号化の鍵として用いることを特徴とする請求項１３ないし請求項１７のいずれか記載のファイル暗号化システムの制御方法。

【請求項１９】　前記暗号化手段は、復号を許可するユーザーのリストの文字列をハッシュ関数によって圧縮することによって作成した数値で暗号化の鍵を暗号化し、該暗号化鍵を暗号化ファイルに付加することを特徴とする請求項１３ないし請求項１７のいずれか記載のファイル暗号化システムの制御方法。

【請求項２０】　少なくとも通信制御機能とファイルへの入出力制御機能を持つオペレーティングシステム（ＯＳ）と、前記ファイルを格納する記憶装置とを具備する第一の情報処理装置と、該第一の情報処理装置とはネットワークで接続され、少なくとも通信制御機能を持つオペレーティングシステム（ＯＳ）と、該ＯＳ上で動作し、前記第一の情報処理装置上の記憶装置からファイルを受信する機能を持つアプリケーションプログラムとを具備する第二の情報処理装置によって構成された暗号ファイル受信システムにおいて、前記第一の情報処理装置が、前記ファイルを暗号化するファイル暗号化手段を有し、前記第二の情報処理装置が、前記アプリケーションプログラムが前記第一の情報処理装置からファイル暗号化手段によって暗号化された前記暗号ファイルを受信する際に、該受信ファイルを復号化してから前記アプリケーションプログラムに渡す受信ファイル復号化手段を有することを特徴とする暗号ファイル受信システム。

【請求項２１】　受信ファイル復号化手段は、前記ファイル暗号化手段によって前記暗号ファイルに復号可能なユーザーのリストが付加されたファイルを受信して、復号化することを特徴とする請求項２０記載の暗号ファイル受信システム。

【請求項２２】　前記受信ファイル復号化手段は、アプリケーションプログラムが前記ＯＳに発行するデータ送受信要求をフックするフック手段によって呼び出されることを特徴とする請求項２０又は請求項２１記載の暗号ファイル受信システム。

【請求項２３】　前記フック手段は、前記第二の情報処理装置のユーザーの身元を確認するユーザー認証手段によって認証されたユーザーが存在する場合にのみ前記受信ファイル復号化手段を呼び出すことを特徴とする請求項２０ないし請求項２２のいずれか記載の暗号ファイル受信システム。

【請求項２４】　前記受信ファイル復号化手段は、前記ユーザー認証手段によって認証されたユーザーが、前記復号可能なユーザーのリストに含まれているか否かを判別する復号権利判別手段によって復号権利があると判断された場合にのみ呼び出されることを特徴とする請求項２０ないし請求項２３のいずれか記載の暗号ファイル受信システム。

【請求項２５】　前記暗号ファイルに、復号可能なユーザーのリストを暗号化して付加したファイルを受信して復号化することを特徴とする請求項２０記載の暗号ファイル受信システム。

【請求項２６】　前記第一の情報処理装置上の記憶装置には、前記ファイル暗号化手段によって予め暗号化されたファイルが格納されていることを特徴とする請求項２０ないし請求項２５のいずれか記載の暗号ファイル受信システム。

【請求項２７】　前記フック手段は、前記アプリケーションプログラムが、前記第一の情報処理装置へ送信したファイル転送要求コマンドに対する受信ファイルのみを復号化の対象として処理することを特徴とする請求項２０ないし請求項２５のいずれか記載の暗号ファイル受信システム。

【請求項２８】　少なくとも通信制御機能とファイルへの入出力制御機能を持つオペレーティングシステム（ＯＳ）と、前記ファイルを暗号化するファイル暗号化手段と、前記暗号化ファイルを格納する記憶装置とを具備する第一の情報処理装置と、該第一の情報処理装置とはネットワークで接続され、少なくとも通信制御機能を持つオペレーティングシステム（ＯＳ）と、該ＯＳ上で動作し、前記第一の情報処理装置上の記憶装置から前記暗号ファイルを受信する機能を持つアプリケーションプログラムとを具備する第二の情報処理装置から構成された暗号ファイル受信システムにおいて、前記ファイル暗号化手段は、前記アプリケーションプログラムが解釈し処理する文法に従って記述された平文ファイルのデータ暗号化において、前記文法における前記アプリケーションプログラム処理時に処理対象外として無視する注釈文とするための記述子を、前記暗号化後のデータに付加したものを暗号ファイルとするものであり、前記アプリケーションプログラムが前記暗号ファイルを受信したとき、前記記述子が付加されて注釈文となっているデータ部分のみを取り出して復号化する受信ファイル復号化手段を設

けたことを特徴とする暗号ファイル受信システム。

【請求項２９】　少なくとも通信制御機能とファイルへの入出力制御機能を持つオペレーティングシステム（ＯＳ）と、前記ファイルを暗号化するファイル暗号化手段と、前記暗号化ファイルを格納する記憶装置とを具備する第一の情報処理装置と、該第一の情報処理装置とはネットワークで接続され、少なくとも通信制御機能を持つオペレーティングシステム（ＯＳ）を具備する第二の情報処理装置と、前記ネットワークで接続され、前記第二の情報処理装置が前記第一の情報処理装置へ送信するデータや、反対に前記第二の情報処理装置が前記第一の情報処理装置から受信するデータを中継する機能と、該受信データをキャッシュするキャッシュ機能とを具備する第三の情報処理装置とから構成される暗号化ファイル受信システムにおいて、前記第二の情報処理装置が、前記暗号ファイルの転送要求を出すと、該暗号ファイルが前記第一の情報処理装置から前記第三の情報処理装置を経由して前記第二の情報処理装置へ転送されると共に、該暗号ファイルは前記キャッシュ機能によってキャッシュされ、前記第二の情報処理装置が、再度同じ暗号ファイルの転送要求を出すと、前記キャッシュ機能によりキャッシュされた暗号ファイルが前記第三の情報処理装置から前記第二の情報処理装置へ転送されることを特徴とする暗号ファイル受信システム。

【請求項３０】　少なくとも通信制御機能とファイルへの入出力制御機能を持つオペレーティングシステム（ＯＳ）と、前記ファイルを格納する記憶装置とを具備する第一の情報処理装置と、該第一の情報処理装置とはネットワークで接続され、少なくとも通信制御機能を持つオペレーティングシステム（ＯＳ）と、該ＯＳ上で動作し、前記第一の情報処理装置上の記憶装置からファイルを受信する機能を持つアプリケーションプログラムとを具備する第二の情報処理装置によって構成された暗号ファイル受信システムの制御方法において、前記アプリケーションプログラムが前記第一の情報処理装置から前記暗号ファイルを受信する際に、該受信ファイルを復号化してから前記アプリケーションプログラムに渡すようにしたことを特徴とする暗号ファイル受信システムの制御方法。

【請求項３１】　前記ファイル暗号化手段によって、前記暗号ファイルに復号可能なユーザーのリストが付加されたファイルを受信して、復号化するようにしたことを特徴とする請求項３０記載の暗号ファイル受信システムの制御方法。

【請求項３２】　前記受信ファイル復号化手段を、アプリケーションプログラムが前記ＯＳに発行するデータ送受信要求をフックするフック手段によって呼び出すことを特徴とする請求項３０記載の暗号ファイル受信システムの制御方法。

【請求項３３】　前記フック手段は、前記第二の情報処理装置のユーザーの身元を確認するユーザー認証手段によって認証されたユーザーが存在する場合にのみ前記受信ファイル復号化手段を呼び出すことを特徴とする請求項３０ないし請求項３２のいずれか記載の暗号ファイル受信システムの制御方法。

【請求項３４】　前記受信ファイル復号化手段を、前記ユーザー認証手段によって認証されたユーザーが、前記復号可能なユーザーのリストに含まれているか否かを判別する復号権利判別手段によって復号権利があると判断された場合にのみ呼び出すことを特徴とする請求項３０ないし請求項３３のいずれか記載の暗号ファイル受信システムの制御方法。

【請求項３５】　前記暗号ファイルに、復号可能なユーザーのリストを暗号化して付加したファイルを受信して復号化することを特徴とする請求項３０記載の暗号ファイル受信システムの制御方法。

【請求項３６】　前記第一の情報処理装置上の記憶装置に、前記ファイル暗号化手段によって予め暗号化されたファイルを格納することを特徴とする請求項３０ないし請求項３５のいずれか記載の暗号ファイル受信システムの制御方法。

【請求項３７】　前記フック手段は、前記アプリケーションプログラムが、前記第一の情報処理装置へ送信したファイル転送要求コマンドに対する受信ファイルのみを復号化の対象として処理することを特徴とする請求項３０ないし請求項３５記載の暗号ファイル受信システムの制御方法。

【請求項３８】　少なくとも通信制御機能とファイルへの入出力制御機能を持つオペレーティングシステム（ＯＳ）と、前記ファイルを暗号化するファイル暗号化手段と、前記暗号化ファイルを格納する記憶装置とを具備する第一の情報処理装置と、該第一の情報処理装置とはネットワークで接続され、少なくとも通信制御機能を持つオペレーティングシステム（ＯＳ）と、該ＯＳ上で動作し、前記第一の情報処理装置上の記憶装置から前記暗号ファイルを受信する機能を持つアプリケーションプログラムとを具備する第二の情報処理装置から構成された暗号ファイル受信システムの制御方法において、前記ファイル暗号化手段は、前記アプリケーションプログラムが解釈し処理する文法に従って記述された平文ファイルのデータ暗号化において、前記文法における前記アプリケーションプログラム処理時に処理対象外として無視する注釈文とするための記述子を、前記暗号化後のデータに付加したものを暗号ファイルとし、前記アプリケーションプログラムが前記暗号ファイルを受信したとき、前記記述子が付加されて注釈文となっているデータ部分のみを取り出して復号化することを特徴とする暗号ファイル受信システムの制御方法。

【請求項３９】　少なくとも通信制御機能とファイルへの入出力制御機能を持つオペレーティングシステム（Ｏ

Ｓ）と、前記ファイルを暗号化するファイル暗号化手段と、前記暗号化ファイルを格納する記憶装置とを具備する第一の情報処理装置と、該第一の情報処理装置とはネットワークで接続され、少なくとも通信制御機能を持つオペレーティングシステム（ＯＳ）を具備する第二の情報処理装置と、前記ネットワークで接続され、前記第二の情報処理装置が前記第一の情報処理装置へ送信するデータや、反対に前記第二の情報処理装置が前記第一の情報処理装置から受信するデータを中継する機能と、該受信データをキャッシュするキャッシュ機能とを具備する第三の情報処理装置とから構成される暗号ファイル受信システムの制御装置において、前記第二の情報処理装置が前記暗号ファイルの転送要求を出すと、該暗号ファイルが前記第一の情報処理装置から前記第三の情報処理装置を経由して前記第二の情報処理装置へ転送されると共に、該暗号ファイルは前記キャッシュ機能によってキャッシュされ、前記第二の情報処理装置が、再度同じ暗号ファイルの転送要求を出すと、前記キャッシュ機能によりキャッシュされた暗号ファイルを前記第三の情報処理装置から前記第二の情報処理装置へ転送することを特徴とする暗号ファイル受信システムの制御方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、情報処理装置のファイルの暗号化システムおよびその制御方法ならびに通信回線を通じて暗号ファイルを受信する際のデータ受信制御方法に係り、特に、複数のユーザーが復号可能な暗号ファイルを作成あるいは復号化する場合に好適な、共用ファイル暗号システム及びその制御方法ならびにアプリケーションプログラムにとって透過な暗号ファイル受信システム及びその制御方法に関する。

【０００２】

【従来の技術】パーソナルコンピュータなどの情報処理装置が普及するにつれて、各情報処理装置毎の磁気記憶装置上に保管していたユーザーのプログラムファイル、データファイルなどを一箇所の大容量磁気記憶装置に保管し、各情報処理装置のユーザーがファイルを共用する、といった使い方が行われるようになってきた。一般的には、ファイルサーバプログラムが搭載されたサーバ情報処理装置と該サーバ情報処理装置へアクセスするためのクライアントプログラムが搭載された複数のクライアント情報処理装置をネットワークで接続し、サーバ情報処理装置上のファイルを各クライアント情報処理装置からアクセスし利用する。このようなファイル共用システムは、従来は企業等の内部で閉じていたが、インターネットと呼ばれる世界的規模のネットワークの普及と共に、このインターネットと企業内ネットワークを接続し企業外からもアクセス可能とするケースが増大している。上述のようなファイル共用システムの普及とネットワークの広域化は、ファイルサーバ上の共用ファイルのセキュリティ即ち不正アクセスに対する防御の必要性を従来にも増して高くしている。

【０００３】サーバ情報処理装置上のファイルのセキュリティを保つ方法としては、ファイルサーバプログラムが具備するユーザー認証機構と、該ユーザー認証に基づいた、各ファイルへのアクセス権限設定／確認機構を用いるのが一般的である。しかし、この方式ではネットワーク上を流れるデータの暗号化が行われないため、盗聴による不正アクセスが可能である。そこで、より高いセキュリティが必要な場合には、格納しているファイル或いはネットワーク上を流れるファイルのデータを秘密鍵や公開鍵を用いて暗号化するのが一般的である。これらの暗号化方法については、「データ保護と暗号化の研究」一松信監修、日本経済新聞社（昭５８）などで詳しく開示されている。

【０００４】また、前記インターネットにおいては、情報発信機能を持つＷＷＷ（Ｗｏｒｌｄ－Ｗｉｄｅ　Ｗｅｂ）サーバを利用し、ハイパーテキストと呼ばれる形式の文書ファイルで情報を提供するのが一般的となっている。前記ハイパーテキストを解釈し、グラフィカルに表示するソフトウェアはブラウザと呼ばれ、広く普及している。前記ハイパーテキストの言語仕様としては、ＨＴＭＬ（ＨｙｐｅｒＴｅｘｔＭａｒｋｕｐ　Ｌａｎｇｕａｇｅ）が一般に用いられており、この言語仕様やＨＴＭＬで記述されたファイルを表示するブラウザについては、「ＵＮＩＸ　ＭＡＧＡＺＩＮＥ」（１９９４．１０）ｐｐ．５２－６０などで詳しく述べられている。なお、殆どのＨＴＭＬファイルは他のＨＴＭＬファイルやイメージデータファイルとリンクしており、前記ブラウザはＨＴＭＬファイルを受信した後で、それがリンクしているイメージデータファイルを受信し、該ＨＴＭＬファイルの内容と合わせて同一画面に表示する。これらＨＴＭＬファイルとそれにリンクされたイメージデータファイルを総括してページと呼ぶ。

【０００５】上記ＷＷＷサーバは、ＨＴＴＰ（Ｈｙｐｅｒｔｅｘｔ　Ｔｒａｎｓｆｅｒ　Ｐｒｏｔｏｃｏｌ）という通信規約に基づいて、上記ＨＴＭＬファイルやイメージデータをブラウザに送信する。この時、インターネット上を流れるデータは暗号化されていないため、ＷＷＷサーバに機密情報を含むページを格納することはセキュリティ上問題がある。インターネット上を流れるデータの暗号化手段としては、例えばＳｅｃｕｒｅＨＴＴＰやＳＳＬ（Ｓｅｃｕｒｅ　Ｓｏｃｋｅｔ　Ｌａｙｅｒ）などが挙げられる。また、これらの技術については、「日経コミュニケーション」日経ＢＰ社（１９９５．１２．４　ｐ７５－８０）などに記載されている。

【０００６】

【発明が解決しようとする課題】上述の暗号化は不正アクセスに対する防御ならびにデータの盗聴や改ざんに対する防御という観点からは十分ではあるが、複数のユー

ザーが同一のファイルを安全にアクセスできるようにするためには、ユーザー毎に異なる鍵を用いる必要があり鍵の管理が困難であるという課題がある。また、格納されているファイルを暗号化する場合には、鍵毎に暗号化ファイルを作成する必要があり、記憶装置の記憶容量を多く消費してしまうという課題がある。また、通信時にネットワーク上のデータを暗号化する方法においては、暗号/復号処理のためアクセス速度が低下してしまうという課題もある。さらに、通信時にネットワーク上のデータを暗号化する方法においては、暗号化処理のためアクセス速度が低下してしまうという課題と、WWWサーバが機密情報を平文ファイルで管理しているため、該平文ファイルを管理する情報処理装置に直接アクセスすることで機密情報が漏れてしまうという課題とがある。

【０００７】また、ファイル共用システムの場合、クライアント情報処理装置が必ずしも復号化ソフトウェアプログラムを具備しているとは限らない。暗号化されたファイルを復号化ソフトウェアが具備されていない情報処理装置上のアプリケーションプログラムで読み出した場合、該情報処理装置のモニタ画面上に意味のない文字や記号が表示されたり、該アプリケーションプログラムが異常動作することがあるという課題がある。

【０００８】また、格納するファイルを暗号化する方法では、ユーザーが暗号化操作を忘れてしまい、不正なアクセスから防御すべきファイルを平文で格納してしまうという課題がある。

【０００９】また、インターネット上で上記ブラウザを具備するクライアント情報処理装置が必ずしも復号化ソフトウェアプログラムを具備しているとは限らない。暗号化されたファイルを復号化ソフトウェアが具備されていない情報処理装置上のブラウザで読み出した場合、該情報処理装置のモニタ画面上に意味のない文字や記号が表示されたり、該ブラウザプログラムが異常動作することがあるという課題がある。

【００１０】そこで、本発明の目的は、複数のユーザーが同一のファイルを利用する場合にも暗号化ファイルが一個で済み且つ安全なファイル暗号化システムおよびその制御方法を提供することにある。

【００１１】また、本発明の他の目的は、複数のユーザーが同一のディレクトリ内のファイルを暗号化する場合に、復号化を許可するユーザーをユーザー毎に設定可能なファイル暗号化システムおよびその制御方法を提供することにある。

【００１２】また本発明の他の目的は、復号化ソフトウェアを具備していないクライアント情報処理装置において暗号化されたファイルを読み出した場合にも、異常動作することのないファイル暗号化システムおよびその制御方法を提供することにある。

【００１３】また、本発明の他の目的は、ユーザーが暗号化を忘れることのないファイル暗号化システムおよびその制御方法を提供することにある。

【００１４】さらに、本発明の目的は、インターネット上で複数のユーザーが同一のページをWWWサーバから受信する場合にも暗号ファイルが１ファイルにつき一個で済み、且つ安全な暗号ファイル受信システム及びその制御方法を提供することにある。

【００１５】また、本発明の他の目的は、復号化ソフトウェアを具備していないクライアント情報処理装置において暗号化されたファイルを読み出した場合にも、異常動作することのない暗号ファイル受信システム及びその制御方法を提供することにある。

【００１６】さらに、本発明の他の目的は、受信した暗号ファイルの復号化処理を、ユーザーがファイル毎に明示的に指示する必要のない暗号ファイル受信システム及びその制御方法を提供することにある。

【００１７】

【課題を解決するための手段】上記目的は、少なくともファイルへの入出力制御機能を持つオペレーティングシステム（ＯＳ）と、前記ファイルを格納する記憶装置とを具備する情報処理装置において、暗号化を行う対象ディレクトリを予めユーザー毎に複数設定可能な暗号化ディレクトリ記憶手段と、前記記憶手段の記憶データを暗号化する手段と、前記ＯＳ上で動作するアプリケーションプログラムから前記ディレクトリ内のファイルへの書き込みデータを暗号化する暗号化手段とを設けることによって達成される。

【００１８】また、上記他の目的は、少なくともファイルへの入出力制御機能を持つオペレーティングシステム（ＯＳ）と、前記ファイルを格納する記憶装置と、前記ファイルを暗号化するファイル暗号化手段と、前記ファイルを復号化する復号化手段とを具備する情報処理装置において、前記ファイル暗号化手段は、アプリケーションプログラムが解釈し処理する文法に従って記述された平文ファイルのデータ暗号化において、前記文法における前記アプリケーションプログラム処理時に処理対象外として無視する注釈文とするための記述子を前記暗号化後のデータに付加し暗号ファイルとして格納し、前記ファイル復号化手段は、前記暗号ファイルのデータ復号化において、前記注釈文のための記述子によって注釈文とされたデータ部分を復号化し、暗号化前の平文ファイルのデータとして格納することによって達成される。

【００１９】また、上記他の目的は、前記アプリケーションプログラムから前記ＯＳに発行されるファイル操作要求をフックするフック手段を設け、前記ファイル操作要求発行時に自動的に暗号化あるいは復号化を行う手段を設けることによって達成される。

【００２０】上記目的は、少なくとも通信制御機能とファイルへの入出力制御機能を持つオペレーティングシステム（ＯＳ）と、前記ファイルを格納する記憶装置とを具備する第一の情報処理装置と、該第一の情報処理装置

とはネットワークで接続され、少なくとも前記第一の情報処理装置上の記憶装置からファイルを受信するための通信制御機能を持つオペレーティングシステム（ＯＳ）を具備する第二の情報処理装置から構成されるシステムにおいて、前記ファイルを暗号化して該暗号ファイルに情報開示先となるユーザーのリストを添付するファイル暗号化手段と、前記第二の情報処理装置を利用するユーザーが、前記記憶装置に登録された前記暗号ファイルを受信したときに、該暗号ファイルを復号化する受信ファイル復号化手段と、前記ユーザーが前記ユーザーリストに含まれている場合には、前記受信ファイル復号化手段を呼び出し、含まれていない場合には復号化を行わないよう制御する復号権利判別手段とを設けることによって達成される。

【００２１】また、上記他の目的は、少なくともファイルへの入出力制御機能を持つオペレーティングシステム（ＯＳ）と、前記ファイルを格納する記憶装置と、前記ファイルを暗号化するファイル暗号化手段とを具備する第一の情報処理装置において、前記ファイル暗号化手段は、アプリケーションプログラムが解釈し処理する文法に従って記述された平文ファイルのデータ暗号化において、前記文法における前記アプリケーションプログラム処理時に処理対象外として無視する注釈文とするための記述子を前記暗号化後のデータに付加し暗号ファイルとして格納し、前記受信ファイル復号化手段は、前記暗号ファイルのデータ復号化において、前記注釈文のための記述子によって注釈文とされたデータ部分を復号化し、暗号化前の平文ファイルのデータとしてアプリケーションプログラムに渡すことによって達成される。

【００２２】また、上記他の目的は、前記第二の情報処理装置において、前記アプリケーションプログラムから前記ＯＳに発行されるデータ送受信要求をフックするフック手段を設け、該フック手段により前記アプリケーションプログラムより先に受信データをチェックし、該受信データが前記暗号ファイルのデータであれば復号化を行い、前記アプリケーションプログラムに渡すことによって達成される。

【００２３】

【発明の実施の形態】以下、本発明の一実施例を図を用いて説明する。

【００２４】まず、全体のシステム構成を説明する。図１は、本発明の共用ファイル暗号システムの一構成例である。１１０はクライアント情報処理装置、１２０はサーバ情報処理装置である。サーバ情報処理装置１２０には、中央処理装置（ＣＰＵ）１ｂ、メモリ２ｂ、ＬＡＮコントローラ３ｂ、ディスクコントローラ４ｂ、磁気ディスク５ｂが具備されている。前記サーバ情報処理装置１２０の起動時には、オペレーティングシステム（ＯＳ）２０ｂおよびファイルサーバプログラム２５が、ディスクコントローラ４ｂを介して磁気ディスク５ｂからメモリ２ｂ上にロードされる。クライアント情報処理装置１１０には、ＣＰＵ１ａ、メモリ２ａ、ＬＡＮコントローラ３ａ、ディスクコントローラ４ａ、磁気ディスク５ａ、ＩＣカードコントローラ６がそれぞれ具備されている。前記クライアント情報処理装置１１０の起動時には、ＯＳ２０ａ、クライアントプログラム２４、共用ファイル暗号プログラム２１、ファイルＩ／Ｏフックプログラム２２が、前記ディスクコントローラ４ａを介して前記磁気ディスク５ａから前記メモリ２ａ上にロードされる。また、アプリケーションプログラム２３は、前記クライアント情報処理装置１１０のユーザー（以下、単にユーザーと呼ぶ）の指示によって前記ディスクコントローラ４ａを介して前記磁気ディスク５ａから前記メモリ２ａ上にロードされる。前記ファイルＩ／Ｏフックプログラム２２は、アプリケーションプログラム２３が前記ＯＳ２０ａに発行するファイルＩ／Ｏ　ＡＰＩをフック（横取り）し、ファイルデータの暗号化・復号化をバックグラウンドで自動的に行うものである。

【００２５】前記サーバ情報処理装置１２０と前記クライアント情報処理装置１１０は、ローカルエリアネットワーク（ＬＡＮ）１０１を介して相互に接続されており、各々のＬＡＮコントローラ３を介して通信を行う。図１では一台のクライアント情報処理装置１１０を示したが、実際には、同じ構成のクライアント情報処理装置１１０が複数前記ＬＡＮ１０１に接続されており、これらのクライアント情報処理装置１１０間で、前記サーバ情報処理装置１２０の前記磁気ディスク５ｂに格納されたファイルを共用する。ユーザーは前記共用を行うため、前記クライアントプログラム２４を介して前記磁気ディスク５ｂ上の共用ファイルをアクセスする。

【００２６】また、クライアント情報処理装置１１０のＩＣカードコントローラ６は、ＩＣカード１３０が脱着可能となっている。前記ＩＣカード１３０内には、ＣＰＵ１ｃ、不揮発性メモリ７、入出力インタフェース８が具備されている。前記不揮発性メモリ７には、アクセス制御プログラム３０、グループ暗号プログラム２８、ＩＤ情報２９、マスタ鍵３１が書き込まれている。前記不揮発性メモリ７の内容は、前記入出力インタフェース８を介して外部からアクセスすることができる。ただし、不正なアクセスから前記不揮発性メモリ７の記憶内容を保護する為、前記ＣＰＵ１ｃによるユーザー認証制御が必ず介在する仕組みとなっている。

【００２７】図２７は、前記共用ファイル暗号プログラム２１と前記ファイルＩ／Ｏフックプログラム２２を構成するプログラムルーチンを示す図である。前記共用ファイル暗号プログラム２１は、前記ＩＣカード１３０へのログイン、ログアウトを行うログイン制御ルーチン１０００、ログアウト制御ルーチン１１００、ユーザーがファイルの暗号化をその都度明示的に指示し暗号化する手動ファイル暗号化ルーチン２０００、前記手動ファイ

ル暗号化をＨＴＭＬ（Ｈｙｐｅｒ　Ｔｅｘｔ　Ｍａｒｋｕｐ　Ｌａｎｇｕａｇｅ）で記述した共用ファイル向けにアレンジしたＨＴＭＬファイル暗号化ルーチン２１００から構成される。前記ファイルＩ／Ｏフックプログラム２２は、ファイルオープンフックルーチン１２００、ファイルクリエイトフックルーチン１３００、ファイルリードフックルーチン１４００、ファイルライトフックルーチン１５００、ファイルクローズフックルーチン１６００、ファイルリネームフックルーチン１７００から構成される。

【００２８】図２は、前記ＩＤ情報２９の内容の一例を示す図である。ここでは、カテゴリ毎の内容を示すデータとそのコードを１０組まで登録できるようになっている。実際に前記不揮発性メモリ７に記憶するのは、データとコードの値だけである。

【００２９】次に、上記グループ暗号プログラム２８の動作について説明する。グループ暗号では、復号化を許す相手の宛先リストを作成し、該宛先リストをハッシュ関数に入力することによって得られるハッシュ値を、暗号化・復号化の鍵に用いる。前記鍵をグループ鍵５１と呼ぶ。本実施例における宛先リストは、前記ＩＤ情報２９を条件式で連結した形で表現する。具体的には、「カテゴリ番号」「データコード識別子」「演算子」「データ」を一組とし、カンマ（，）で区切って並べる。このカンマは論理和の意味を持つ。演算子としては、等しい（＝）の他、大小（＞、＜）や論理積（＾）を用いることができる。例えば、図２で示した前記ＩＤ情報２９の定義を用いた場合、

「システム研究所の第四部の主任以上と暗号太郎」

という宛先に対する宛先リストは、

「５Ｃ＝６７６＾６Ｃ＝０４＾８Ｃ＞＝５，２Ｄ＝暗号太郎」

となる。

【００３０】図３は、前記宛先リストのハッシュ値を計算し前記グループ鍵５１を作成する手順の一例を示す図である。３１はマスタ鍵、４１は宛先リスト、５１はグループ鍵、３２はハッシュ関数である。まず、前記ＩＤ情報を元に作成した宛先リストのデータ長が８の倍数のバイト数になっていない場合には不足分を空白文字で補い更に８バイトの乱数を付加する。宛先リストが丁度８の倍数であった場合には単に８バイトの乱数を付加する。８の倍数とした宛先リストを以下では単に宛先リスト４１と呼ぶ。このように乱数を付加することによって、同じ宛先でも毎回異なるグループ鍵が生成されるのでセキュリティを向上することができる。

【００３１】次にハッシュ関数３２の動作を簡単に説明する。まず、前記宛先リスト４１を先頭から８バイトずつブロック暗号方式によって暗号化し、該暗号化の結果と初期値との排他的論理和を取り次の段のブロック暗号の初期値として用いる。この処理を前記宛先リスト４１の最後まで繰り返し、最後の段から出力される暗号化結果と初期値との排他的論理和をグループ鍵５１とする。なお、前記ブロック暗号では、８バイトのマスタ鍵３１と８バイトの初期値を用いて入力データの暗号化を行う。なお、データを暗号化するユーザーと復号化するユーザーは、同じ値の前記マスタ鍵３１を使用する。

【００３２】図４は、前記ＩＣカード１３０内の前記グループ暗号プログラム２８の処理内容を示すフロー図である。グループ暗号プログラム２８では、上述のグループ鍵生成方法を用いて前記宛先リスト４１からグループ鍵５１を作成する。ステップ４０１では、前記ＩＣカード１３０のアクセス制御プログラム３０に対するログインが完了しているかどうかを検査し、完了していなければステップ４１１でエラーコードを設定して本プログラムを終了する。前記ログインが完了している場合には、入力された宛先リスト４１をステップ４０２で解析し、前記ＩＤ情報２９に該当する宛先があるかどうかを検査する。該検査の結果宛先が該当していればステップ４０３を実行し、該当していなければステップ４１２でエラーコードを設定し本プログラムを終了する。ステップ４０３では、図３で説明した方法でグループ鍵を生成する。以上の処理によって、ＩＣカード１３０にログイン済みで且つ前記宛先リストの宛先としてＩＣカード１３０内のＩＤ情報２９に該当するものがある場合のみグループ鍵を生成するように制御することができる。

【００３３】以上説明したグループ鍵５１を用いてデータを暗号化することによって、前記宛先リスト４１に含まれるＩＤ情報２９を持つ者だけが該データの復号化を可能とすることができる。

【００３４】次に、上述のグループ暗号をファイルの暗号に適用した例について説明する。図５は、ファイルの暗号化手順と暗号化後のファイルの構成を示す図である。５０１は暗号化する前の平文ファイル、５０２は暗号化後の暗号文ファイルである。６１は乱数として生成された実行鍵、５０４は前記実行鍵６１を鍵として前記ファイル５０１の内容を暗号化した暗号文データ、５０５は暗号文データ５０４に付加するヘッダ、５０６から５１１は前記ヘッダ５０５の内容であり、５０６は使用した暗号化方式を示す文字列、５０７は前記平文ファイル５０１のファイル名、５０８前記平文ファイル５０１のファイルサイズ、５０９は前記実行鍵６１をグループ鍵５１で暗号化した暗号化実行鍵、５１０は前記宛先リスト４１のサイズ、５１１は前記宛先リスト４１を前記マスタ鍵３１で暗号化した暗号化宛先リストである。前記暗号文ファイル５０２は、図６に示すように上記手順と全く逆の手順で復号化することができる。上述のファイル暗号化においては、前記グループ鍵５１は前記実行鍵６１を暗号化する為に用い、ファイルデータの暗号化には前記実行鍵６１を用いた。したがって、前記宛先リスト４１を変更する場合には、まず前記実行鍵６１を変

更前の前記宛先リスト４１で生成した前記グループ鍵５１で復号化し、次に変更後の前記宛先リスト４１から生成した前記グループ鍵５１で再暗号化すればよい。これにより、前記宛先リスト４１の変更時にも前記実行鍵６１に比べデータ量の多いファイルデータは再暗号化が不要とすることができ、変更処理を高速に行うことができる。

【００３５】次に、上記ファイルの暗号化を用いてユーザーが操作するファイルを自動的に暗号化・復号化する機能について説明する。ここで用いるファイルシステムは、最近のＯＳが具備する一般的なものであり、ツリー状の階層構造を持つ。ツリーの節の部分はディレクトリと呼ばれ、各ディレクトリに存在するファイルはディレクトリファイルと呼ばれるファイルによって管理される。ファイルの物理的な格納場所は、前記クライアント情報処理装置１１０の前記磁気ディスク５ａあるいは前記サーバ情報処理装置１２０の前記磁気ディスク５ｂのどちらでもよい。

【００３６】図７は、ユーザーから見た論理的なディレクトリの構成を示す図である。本実施例のファイルの自動暗号化、復号化を導入していない場合は、前記論理的なディレクトリ構成と物理的なディレクトリ構成は同一である。図８は、ファイルの自動暗号化、復号化を導入した後の物理的なディレクトリ構成を示す図である。ユーザーがファイルの自動暗号化、復号化を行うディレクトリには、ファイル名としてユーザー名とその末尾に「．ｄｅｉ」を付加したファイル名で宛先リストファイル９０１を作成する。また、全ユーザーに共通な宛先を設定して暗号化を行う場合には、「ｄｅｆａｕｌｔ．ｄｅｉ」というファイル名で宛先リストファイル９０１を作成する。このように、前記宛先リストファイル９０１を作成しておくと、前記宛先リストファイル９０１に記載された宛先リストを宛先としたファイルの自動暗号化が行われ、連番とその末尾に「．ｓｓｆ」を付加したファイル名で格納される。例えば、図７のファイル「ｏｓａｋａ．ｄｏｃ」は「０００１．ｓｓｆ」として格納される。また前記自動暗号化が行われたディレクトリには、「ｆｉｌｅｉｎｆｏ．ｄｅｉ」という名称の暗号ファイル情報ファイル９０２が作成される。

【００３７】図９は、前記宛先リストファイル９０１、前記暗号ファイル情報ファイル９０２の内容の一例を示す図である。各ユーザーの宛先リストファイル９０１は、該ユーザーを宛先としたグループ暗号で暗号化している。また、「ｄｅｆａｕｌｔ．ｄｅｉ」は、前記マスタ鍵３１を用いて暗号化している。これによって、宛先リストファイルの不正な読み出しや改竄を防止できる。一方、暗号ファイル情報ファイル９０２には、暗号化後のファイル名、暗号化前の平文ファイル名、暗号化前の平文ファイルのサイズが書き込まれている。上述のファイル名称はあくまでも一例であり他の名称を用いてもよい。

【００３８】以下、図８に示したような物理的構造を持つファイルを図７に示した論理的ファイル構造として扱うための処理プログラムの詳細を説明する。

【００３９】まず、共用ファイル暗号プログラム２１内のログイン処理ルーチン１０００、ログアウト処理ルーチン１１００について説明する。図１０は、前記ＩＣカード１３０へのログイン処理ルーチン１０００の処理フローを示す図である。まず、ステップ１００１でユーザーがＩＤとパスワードの入力を行い、ステップ１００２では前記ユーザーＩＤと前記パスワードを用いて前記ＩＣカード１３０にログインを試み、ステップ１００３で前記ログインが成功したかどうかを判定する。前記判定の結果、失敗の場合はステップ１００９を実行し、３回続けて失敗するまではステップ１００１以下を繰り返すよう制御する。ログインに成功すると、ステップ１００４でイベント待ち状態となる。イベントが到着すると、ログアウト要求かどうかを調べ、ログアウト要求ならばステップ１００８でログアウト処理を行い終了する。前記イベントがログアウト要求でなければステップ１００６でタイマイベントかどうかを調べ、タイマイベントでなければステップ１００４に戻る。タイマイベントであった場合には、ＩＣカード１３０が前記クライアント情報処理装置１１０に接続されているかどうかを調べ、接続されていればステップ１００４に戻り、接続されていない場合は、ステップ１００８のログアウト処理を実行し、終了する。このように、タイマイベントを用いて定期的に前記ＩＣカード１３０の接続を調べることによって前記ＩＣカード１３０を抜いた際のログアウト処理を自動化することができる。

【００４０】図１１は、前記ＩＣカード１３０からのログアウト処理ルーチン１１００の処理フローを示す図である。まず、ステップ１１０１でオープン中の暗号ファイルがあるかどうかを調べ、あればステップ１１０２でユーザーにログアウトを中止するか否かを尋ねる。ログアウト処理を中止する場合には、本ルーチンを終了する。中止しない場合には、ステップ１１０３でオープン中の暗号ファイルのクローズ処理を行い、処理を継続する。ステップ１１０４では、ＩＣカード１３０がクライアント情報処理装置１１０に接続されているかどうかを調べ、接続されていればステップ１１０５でＩＣカード１３０に対してログアウトコマンドを送出し処理を終了する。接続されていなければそのまま処理を終了する。

【００４１】次に、前記ファイルＩ／Ｏフックプログラム２２内の各ルーチンの処理について説明する。まず、ファイルのデータ入出力とその管理方法から説明する。

【００４２】図１８は、本実施例のファイル暗号化・復号化におけるデータの入出力方法を示す図である。前記暗号文ファイル５０２の前記暗号文データ５０４は、暗号文ブロックバッファ８２のサイズ単位で前記暗号文ブ

ロックバッファ８２との間で読み出し・書き込みが行われる。前記暗号文ブロックバッファ８２のデータは復号化し平文ブロックバッファ８１に転送される。また、前記平文ブロックバッファ８１のデータは暗号化し前記暗号文ブロックバッファ８２に転送される。

【００４３】図１９は、オープン中の暗号ファイルを管理するためのファイルハンドルテーブル９０である。ハンドル番号１９０２、オープンモード１９０３、平文ファイルポインタ１９０４、暗号ファイルポインタ１９０５、バッファポインタ１９０６、更新フラグ１９０７、宛先リスト４１の各情報を、オープン中の各ファイル毎にポインタ１９０１を用いてリスト構造で記憶している。ここでハンドル番号１９０２は、ファイルをオープンした際に前記ＯＳ２０から与えられる管理番号、オープンモード１９０３とは、読み出し専用、書き込み専用、あるいは読み書き用などのモードを示す値である。前記平文ファイルポインタ１９０４は復号化後の平文ファイル５０１に対するリード・ライトを行う際の起点を示すものであり、平文ファイル５０１の先頭からのバイト数で示される。なお、平文ファイル５０１は、平文ブロックバッファ８１を通して読み書きする論理的なファイルとして存在する。暗号ファイルポインタ１９０５は、暗号文ファイルに対するリード・ライトを行う際の起点を示すものであり、５０２のヘッダ５０５を含む先頭からのバイト数で示される。前記暗号ファイルポインタ１９０５は、ファイルの最後のブロック部分以外ではバッファサイズ単位で増減する。バッファポインタ１９０６は、平文ブロックバッファ８１および暗号文ブロックバッファ８２に転送されているデータの先頭が前記平文ファイル５０１のどの部分に該当するデータであるかを示す値であり、前記平文ファイル５０１の先頭からのバイト数が格納してある。前記バッファポインタ１９０６はバッファサイズの単位でのみ増減する。更新フラグ１９０７は、前記平文ブロックバッファ８１のデータが、ライトＩ／Ｏ操作によって更新されているが暗号文ブロックバッファ８２への転送と暗号文ファイル５０２への書き込みが完了していない状態を示すフラグである。

【００４４】図１２は、前記ファイルオープンフックルーチン１２００の処理フローを示す図である。まず、ステップ１２８１で、フックしたファイルのディレクトリに前記暗号ファイル情報ファイル９０２が存在するか否かを調べ、存在すれば前記暗号ファイル情報ファイル９０２の内容を参照し、フックしたファイルのファイル名が登録されているか否かを調べる。登録されていれば、ステップ１２８２で対応する暗号文ファイル５０２のファイル名称にファイル名称を変更する。次に、ステップ１２０１で前記ＯＳ２０ａに対しファイルオープン要求を発行する。該オープンが成功したかどうかをステップ１２０２で調べ、失敗した場合は本ルーチンを終了する。次にステップ１２０３で、ＩＣカード１３０へのログインが終了しているかどうかを調べ、ログイン済みの場合は、ステップ１２０７に処理を移す。ログイン済みでなかった場合には、ステップ１２０４で当該ファイルが暗号ファイルであるかどうかを調べ、暗号ファイルでなかった場合は本ルーチンの処理を終了する。暗号ファイルの場合には、ステップ１２０５で前記ログイン処理ルーチン１０００を呼び出すことによってＩＣカード１３０へのログイン処理を行う。ステップ１２０６では、前記ログイン処理が成功したかどうかを調べ、成功しなかった場合はステップ１２１５でメッセージ２　１２５２を表示して本ルーチンを終了する。ログインに成功した場合は、ステップ１２０７で当該ファイルの復号権限があるか、即ち宛先にログイン中のユーザーが含まれているかどうかを判定する。前記判定は暗号ファイルの前記ヘッダ５０５の前記暗号化宛先リスト５１１を復号化することによって前記宛先リスト４１を取得し、該宛先リスト４１を前記ＩＣカード１３０の前記グループ暗号プログラム２８に入力し、グループ鍵５１が出力されるかエラーコードが出力されるかで行う。復号権限がない場合には、ステップ１２１５でメッセージ２　１２５２を表示して本ルーチンを終了する。復号権限がある場合にはステップ１２０９において、当該ユーザーまたは共通の前記宛先リストファイル９０１が当該ディレクトリに存在するかどうかを調べ、存在すれば暗号化対象ディレクトリであると判定する。暗号化対象ディレクトリでなければステップ１２１６を実行し、フックしているファイルが暗号ファイルであるかどうかを調べ、暗号ファイルでなければ本ルーチンを終了する。暗号ファイルの場合は、ステップ１２１７でメッセージ３１２５３を表示し、処理を中断するか否かをユーザーに問い合わせる。上記ステップ１２０９で暗号化対象ディレクトリと判定した場合は、ステップ１２１０でフックしているファイルが暗号ファイルかどうかを調べ、暗号ファイルでなかった場合には、ステップ１２１１でメッセージ１　１２５１を表示し、処理を中断するか否かをユーザーに問い合わせる。ステップ１２１０の判定で、暗号ファイルであった場合には、ステップ１２１８で当該ファイルの宛先リスト４１と当該ディレクトリの宛先リストファイル９０１に書かれた宛先リスト４１の宛先が同じか異なるかを調べ、異なっていた場合には、ステップ１２１９でメッセージ４　１２５４を表示し、処理を中断するか否かをユーザーに問い合わせる。前記ステップ１２１８で宛先が同じと判定した場合には、ステップ１２２０でファイルハンドルテーブル９０への登録を行い、本ルーチンを終了する。また、前記ステップ１２１７、１２１１、１２１９でのメッセージ表示に対し、ユーザーが処理の中止を選択したかどうかはステップ１２１２で判定し、中止の場合はステップ１２１３でファイルをクローズし、ステップ１２１４でリターンコードにオープン

エラーを設定し本ルーチンを終了する。前記ステップ１２１２での前記判定が中止でなければ、前記ステップ１２２０を実行し、本ルーチンを終了する。

【００４５】図１３は、ファイルクリエイトフックルーチン１３００の処理フローを示す図である。まず、ステップ１３０１で、ＩＣカード１３０へのログインが終了しているかどうかを調べ、ログイン済みの場合はステップ１３０６に処理を移す。ログイン済みでなかった場合には、ステップ１３０２で当該ユーザーまたは共通の前記宛先リストファイル９０１が当該ディレクトリに存在するかどうかを調べ、存在すれば暗号化対象ディレクトリであると判定する。暗号化対象ディレクトリでなければ、ステップ１３３１でファイルクリエィトを実行し本ルーチンを終了する。暗号化対象ディレクトリの場合は、ステップ１３０３でメッセージ５　１３５５を表示し、ステップ１３０４で前記ログイン処理ルーチン１０００を呼び出すことによって前記ＩＣカード１３０へのログイン処理を行う。ステップ１３０５では、前記ログイン処理が成功したかどうかを調べ、成功しなかった場合はステップ１３４１でメッセージ６　１３５６を表示してファイルクリエイトを中止するか否かユーザーに問い合わせる。ステップ１３４２でユーザーの選択を判定し、中止の場合はステップ１３４３でクリエイトエラーを設定し本ルーチンを終了する。中止でない場合は、ステップ１３２０でファイルクリエイトを実行し本ルーチンを終了する。一方、ステップ１３０５でログインに成功した場合は、ステップ１３０６で、当該ディレクトリに当該ユーザーまたは共通の前記宛先リストファイル９０１が存在するかどうかを調べ、存在すれば暗号化対象ディレクトリであると判定する。暗号化対象ディレクトリでなければステップ１３２０でファイルクリエイトを実行し、本ルーチンを終了する。暗号化対象ディレクトリである場合は、ステップ１３０７でファイル名称を、連番とその末尾に「．ｓｓｆ」を付加したファイル名に置き換え、ステップ１３０８でファイルクリエイトを実行する。ステップ１３０９では、前記ファイルクリエィトが成功したかどうかを調べ、失敗ならば本ルーチンを終了し、成功ならばステップ１３１０以下を実行する。ステップ１３１０では、暗号ファイルの前記ヘッダ５０５を生成し、ステップ１３０８で作成したファイルの先頭部分に書き込む。続くステップ１３１１では前記暗号ファイル情報ファイル９０２への登録を行い、さらにステップ１３１２でファイルハンドルテーブル９０への登録を行い本ルーチンを終了する。

【００４６】図１４は、リードフックルーチン１４００の処理フローを示す図である。まず、ステップ１４０１でリードＩ／Ｏのパラメータに含まれるファイルハンドル番号が前記ファイルハンドルテーブル９０に既に登録されているかどうかを調べ、登録されていなければ暗号・復号対象のファイルではないのでステップ１４２１で通常のファイルリードＩ／Ｏを実行し、本ルーチンを終了する。ファイルハンドルテーブル９０に登録済みの場合は、ステップ１４０２で、前記ファイルハンドルテーブル９０の前記オープンモード１９０３がリード可になっているかどうかを調べ、リード不可であれば処理が継続できないので、ステップ１４２２でエラーコードを設定して本ルーチンを終了する。次に、ステップ１４０３で前記平文ファイルポインタ１９０４を取得する。ステップ１４０４では、前記平文ファイルポインタ１９０４が、前記バッファポインタ１９０６から始まるバッファの中を指し示しているかどうかを調べる。指し示していればステップ１４１３に処理を移し、指し示していなければステップ１４０５以下を実行する。ステップ１４０５では前記更新フラグ１９０７を参照し、前記平文ブロックバッファ８１中に更新データがあるかどうかを調べ、あればステップ１４０６で実際の暗号文ファイル５０２のファイルポインタをＯＳへのシークコマンドによって前記バッファポインタ１９０６へと移動させる。ステップ１４０７で平文ブロックバッファ８１のデータを暗号化し、暗号文ブロックバッファ８２に転送すると共に、暗号文ファイル５０２に書き戻す。ステップ１４０８では前記更新フラグ１９０７をクリアし、前記平文ブロックバッファ８１中に更新データがないことを示すように設定する。ステップ１４０９で実際の暗号文ファイル５０２のファイルポインタをＯＳへのシークコマンドによって前記暗号ファイルポインタ１９０５へと移動させる。ステップ１４１０では、ブロックサイズ分のデータを前記暗号文ファイル５０２から暗号文ブロックバッファ８２へと読み出す。ステップ１４１１では、前記暗号文ファイルポインタ１９０５、前記バッファポインタ１９０６を更新する。ステップ１４１２では、暗号文ブロックバッファ８２のデータを復号化し平文ブロックバッファ８１に転送する。ステップ１４１３では、フックしたファイルリードＩ／Ｏの要求元アプリケーションプログラム２３が指定した要求バッファに平文ブロックバッファ８１からデータを転送する。ステップ１４１４では、前記転送を行ったバイト数を前記平分ファイルポインタ１９０４に加算する。ステップ１４１５では、フックしたファイルリードＩ／Ｏの要求元が要求したサイズ分のデータを前記要求バッファに転送したかどうかを調べ、未完了ならばステップ１４１０以下を再び実行し、完了ならば本ルーチンを終了する。

【００４７】図１５は、ライトフックルーチン１５００の処理フローを示す図である。まず、ステップ１５０１でフックしたライトＩ／Ｏのパラメータに含まれるファイルハンドル番号が前記ファイルハンドルテーブル９０に既に登録されているかどうかを調べ、登録されていなければ暗号・復号対象のファイルではないのでステップ１５２１で通常のファイルリードＩ／Ｏを実行し、本ルーチンを終了する。ファイルハンドルテーブル９０に登

録済みの場合は、ステップ１５０２で、前記ファイルハンドルテーブル９０の前記オープンモード１９０３が「リードライト可」になっているかどうかを調べ、「リードライト可」でなければステップ１５３１でエラーコードを設定して本ルーチンを終了する。「ライト可」だけではなく「リードライト可」の必要があるのは、暗号化のブロック単位よりも小さなサイズの書込に対しては、不足分のデータをリードした上で暗号化して書き戻すリードモディファイライト処理が必要なためである。前記ファイルオープンフックルーチンで「ライト可」を強制的に「リードライト可」に置き換えてファイルをオープンしておいてもよい。次に、ステップ１５０３で前記平文ファイルポインタ１９０４を取得する。ステップ１５０４では、前記平文ファイルポインタ１９０４が、前記バッファポインタ１９０６から始まるバッファの中を指し示しているかどうかを調べる。指し示していればステップ１５１３に処理を移し、指し示していなければステップ１５０５以下を実行する。ステップ１５０５では前記更新フラグ１９０７を参照し、前記平文ブロックバッファ８１中に更新データがあるかどうかを調べ、更新データがなければステップ１５１０に処理を移し、更新データがあればステップ１５０６で実際の暗号文ファイル５０２のファイルポインタをＯＳへのシークコマンドによって前記バッファポインタ１９０６へと移動させる。ステップ１５０７では、平文ブロックバッファ８１のデータを暗号化し、暗号文ブロックバッファ８２に転送すると共に、暗号文ファイル５０２に書き戻す。ステップ１５０８では前記更新フラグ１９０７をクリアし、前記平文ブロックバッファ８１中に更新データがないことを示すように設定する。ステップ１５０９では、暗号文ファイル５０２のファイルポインタをＯＳへのシークコマンドによって前記暗号ファイルポインタ１９０５が指し示す位置へと移動させる。ステップ１５１０では、フックしたファイルライトＩ／Ｏ要求の書込データ領域が、現在の平文ブロックバッファ８１全体を包含するか否かを調べる。前記調査の結果全体を包含していない場合は、不足分を読み出しマージする必要があるので、ステップ１５１１で暗号文ブロックバッファ８２のサイズ分だけ暗号文ファイル５０２をリードする。リードしたデータは、前記暗号文ブロックバッファ８２に格納すると共に、復号化し前記平文ブロックバッファ８１に格納する。一方、前記ステップ１５１０での調査の結果、全体を包含していれば不足分を読み出す必要がないのでステップ１５１８でステップ１５１１のリードサイズと同じサイズ分だけシーク処理を行う。ステップ１５１２では、暗号文ファイルポインタ１９０５、バッファポインタ１９０６を更新する。ステップ１５１３では、平文ファイルポインタ１９０４の指し示す位置から暗号文ファイルポインタ１９０５の指し示す位置の手前までのデータを、フックしたファイルＩ／Ｏ要求の書込データバッファから平文ブロックバッファ８１に転送する。ステップ１５１４では、平文ブロックバッファ８１の内容が更新されたことを示す更新フラグ１９０７をセットし、ステップ１５１５でフックしたファイルＩ／Ｏライト要求の全データの転送を完了したかどうかを判定し、完了していなければステップ１５０６以下を完了するまで繰り返し実行する。以上の処理によって、平文ブロックバッファ８１は、先読みと先復号化キャッシュとして動作し、且つライトバックキャッシュとして動作するので、暗号化・復号化によるファイルＩ／Ｏ処理速度の低下を少なく抑えることができる。

【００４８】図１６は、クローズフックルーチン１６００の処理フローを示す図である。まず、ステップ１６０１で前記ファイルハンドルテーブル９０に、当該クローズ要求のあったファイルハンドルが登録されているかどうかを調べ、登録されていなければステップ１６０７に処理を移行し、ファイルをクローズして終了する。前記ファイルハンドルテーブル９０に登録済みのファイルであった場合には、前記ファイルハンドルテーブル９０の更新フラグ１９０７を参照し、前記更新フラグ１９０７がセット、即ちファイルに書き戻されていない更新データが平文ブロックバッファ８１に存在するかどうかを調べる。前記更新フラグ１９０７がセットされていた場合には、ステップ１６０３で平文ブロックバッファ８１の内容を暗号化し、ファイルに書き戻す処理を行う。続くステップ１６０４では、前記平文ブロックバッファ８１を、全て０などのデータを書き込むことによってクリアし、ステップ１６０５で前記平文ブロックバッファ８１と前記暗号文ブロックバッファ８２を解放する。ステップ１６０６では、前記ファイルハンドルテーブル９０から当該ファイルのノードを削除し、ステップ１６０７でファイルクローズ要求をＯＳに発行し、本ルーチンを終了する。

【００４９】図１７は、ファイルリネームフックルーチン１７００の処理フローを示す図である。まず、ステップ１７０１でリネームの対象がディレクトリファイルかどうかを判定し、ディレクトリファイルならばステップ１７２１でリネームを実行し、本ルーチンを終了する。ディレクトリファイルでなければ、ステップ１７０２で対象ファイルが暗号ファイルかどうかを判定し、暗号ファイルでなければ前記ステップ１７２１でリネームを実行し、本ルーチンを終了する。続くステップ１７０３では、当該ファイルの復号権限があるか、即ち前記ヘッダ５０５の宛先リスト４１にログイン中のユーザーのＩＤ情報２９とマッチする項目が含まれているかどうかを判定し、復号権限がなければ、ステップ１７１１で権限がない旨のメッセージを表示し、ステップ１７１２でエラーコードを設定し本ルーチンを終了する。復号権限がある場合は、ステップ１７０４で、当該ファイルのヘッダ５０５の平文ファイル名５０７を前記アプリケーション

プログラム２３が指示した新名称に変更し、ステップ１７０５では暗号ファイル情報ファイル９０２に記載された当該ファイルの平文ファイル名称を前記ステップ１７０４と同様に新名称に変更し、本ルーチンを終了する。

【００５０】以上説明したファイルＩ／Ｏフックプログラム２２の各ルーチンの処理によって、所望のディレクトリ内のファイルの暗号化・復号化をアプリケーションプログラム２３からは透過に実現することができる。

【００５１】次に、前記共用ファイル暗号プログラム２１に含まれる手動ファイル暗号化ルーチン２０００について図２０を用いて説明する。このルーチンは、暗号化を行う平文ファイルのファイル名称と、暗号化後の暗号文ファイル名称と宛先リスト４１を入力として動作する。まず、ステップ２００１では、指定された平文ファイルをオープンし、ステップ２００２で仮の名称を付けた一時ファイルを作成する。ステップ２００４では指定された前記宛先リスト４１を前記ＩＣカード１３０のグループ暗号プログラム２８に与えることによってグループ鍵を生成する。前記生成には前記ＩＣカード１３０へのログインが必要であるが，上記の実施例と同様であるのでここでは記述を省略している。ステップ２００５では乱数である前記実行鍵４１を生成し、ステップ２００６で前記ヘッダ５０５を前記一時ファイルの先頭部分に書き込む。ステップ２００９では、平文ファイルからデータを読み出し、ステップ２０１０で該データを暗号化し、ステップ２０１１で該暗号化データを一時ファイルに書き込む。ステップ２０１２では、前記平文ファイルの全データの暗号化を終了したかどうかを判定し、終了するまで前記ステップ２００９、２０１０、２０１１を繰り返し実行するように制御する。ステップ２０１３では、前記平文ファイルと前記一時ファイルの両方をクローズする。ステップ２０１４では、指定された平文ファイル名と暗号文ファイル名が同一かどうかを判定し、同一ならばステップ２０１５で前記平文ファイルを削除する。ステップ２０１６では、前記一時ファイルのファイル名称を指定された暗号文ファイル名称に変更し、本プログラムを終了する。

【００５２】次に、ＨＴＭＬファイルの暗号化を行う実施例について説明する。図２２の２２０１は、平文のＨＴＭＬファイルの一例である。前記ＨＴＭＬファイルは、ＨＴＭＬブラウザで表示すると図２３に示す画面となる。図２４の２４０１は、暗号化したＨＴＭＬファイルの一例であり、リーダ部分２４０２、前記平文のＨＴＭＬファイル２２０１を暗号化した本体部分２４０３、フッタ部分２４０４を合成したものである。リーダ部分２４０２の最後はＨＴＭＬでのコメント文開始を示す「＜！－－」が、フッタ部分２４０４の最初はＨＴＭＬでのコメント文終了を示す「－－＞」が設定されており、前記合成によって本体部分２４０３はコメント文の内容となる。ＨＴＭＬブラウザによる表示において、前記コメント文の内容は表示されないので、前記の暗号化ＨＴＭＬファイル２４０１をＨＴＭＬブラウザで表示した結果は、図２５に示すような画面となる。このように、暗号化した結果をコメント文として埋め込むことによって、復号化プログラムを具備しないクライアント情報処理装置１１０で暗号化ファイルを誤って表示した場合にも、意味のない文字列を画面に表示することを防止することができる。

【００５３】図２１は、上記のＨＴＭＬファイル暗号化を行うＨＴＭＬファイル暗号化プログラムの処理フローを示す図である。ステップ２１０１では、指定された平文のＨＴＭＬファイル２２０１をオープンし、ステップ２１０２で仮の名称を付けた一時ファイルを作成する。ステップ２１０４では指定された前記宛先リスト４１を前記ＩＣカード１３０のグループ暗号プログラム２８に与えることによってグループ鍵を生成する。ステップ２１０５ではリトライカウンタを初期化し、ステップ２１０６では乱数である前記実行鍵４１を生成する。ステップ２１０７では前記ヘッダ５０５を前記一時ファイルの先頭部分に書き込む。ステップ２１０９では、前記リーダ部分２４０２を前記一時ファイルの前記ヘッダ５０５の続きに書き込む。ステップ２１１０では、前記平文のＨＴＭＬファイル２２０１からデータを読み出し、ステップ２１１１で該データを暗号化する。ステップ２１１２では、暗号化後のデータの中に前記コメントの終了文字列「－－＞」がないかどうかを調べ、あればステップ２１３１以下を実行し、なければステップ２１１３で前記一時ファイルに前記暗号化後のデータを書き込む。ステップ２１１４では、前記平文のＨＴＭＬファイル２２０１の全データの暗号化を終了したかどうかを判定し、終了するまで前記ステップ２１１０、２１１１、２１１２、２１１３を繰り返し実行するように制御する。ステップ２１１５では、前記フッタ部分２４０４を前記一時ファイルに書き込み、ステップ２１１６で前記平文のＨＴＭＬファイル２２０１と前記一時ファイルの両方をクローズする。ステップ２１１７では、指定された平文のＨＴＭＬファイル名と暗号化後のＨＴＭＬファイル名が同一かどうかを判定し、同一ならばステップ２１１８で前記平文のＨＴＭＬファイル２２０１を削除する。ステップ２１１９では、前記一時ファイルのファイル名称を指定された暗号化後のＨＴＭＬファイル名称に変更し、本プログラムを終了する。一方、前記ステップ２１１２において、暗号化後のデータの中に前記コメントの終了文字列「－－＞」あると判定された場合は、ステップ２１３１で前記一時ファイルに書き込んだ内容をクリアする。次に、ステップ２１３２で前記リトライカウンタをカウントアップし、ステップ２１３３で前記リトライカウンタが予め設定した上限値（本実施例では１０回とした）を越えていないかを検査し、越えていなければステップ２１０６からの部分を再度実行する。この再実行で

25

は、前記ステップ２１０６で前回とは異なる乱数が実行鍵６１として使われるので、前記ステップ２１１１の暗号化で再び前記コメントの終了文字列「－－＞」が現れる確率はかなり低い。それにも拘わらず、前記上限値を越えてしまった場合は、ステップ２１３４で前記平文のＨＴＭＬファイル２２０１と前記一時ファイルの両方をクローズする。さらにステップ２１１５で前記一時ファイルを削除し、ステップ２１３６でエラーメッセージを表示し、本プログラムを終了する。なお、暗号化したデータはランダムな値となるので、前記コメントの終了文字列「－－＞」が現れる確率は、一文字は１バイト（＝８ビット）で表現されるので「（２の８乗）の３乗」分の１、即ち１６７７７２１６分の１である。言い換えれば約１６Ｍバイトのデータに一回起こる程度ということである。したがって、上述のように実行鍵６１を替えてリトライする制御を組み込んでおけば、実用上全く問題がないことが判る。また、前記コメントの終了文字列「－－＞」が前記本体部分２４０３に出現することを避ける別の手段として、暗号化したデータを１６進表示によるテキスト文字列として格納することもできる。

【００５４】この方法で作成した暗号化ＨＴＭＬファイルを図２６に示す。ただし、この方法では、１バイトのデータを表現するために２文字即ち２バイトが必要であり、データ量が倍になってしまうという問題もある。

【００５５】本実施例では、ＨＴＭＬで記述されたドキュメントファイルについて述べたが、解釈時に無視されるコメント文を記述するための書式を備えた文法であれば同様に本発明を適用することができることは明らかである。

【００５６】

【実施例】以下、本発明をインターネットに用いた場合の実施例を図を用いて説明する。まず、全体のシステム構成を説明する。図１は、本発明の暗号ファイル受信システム及びその制御方法の一構成例である。１１０はクライアント情報処理装置、１２０はサーバ情報処理装置である。

【００５７】サーバ情報処理装置１２０には、中央処理装置（ＣＰＵ）１ｂ、メモリ２ｂ、ＬＡＮコントローラ３ｂ、ディスクコントローラ４ｂ、磁気ディスク５ｂが具備されている。サーバ情報処理装置１２０の起動時には、オペレーティングシステム（ＯＳ）２０ｂと、ＷＷＷ（Ｗｏｒｌｄ－Ｗｉｄｅ　Ｗｅｂ）サーバプログラム２６、および共用ファイル暗号プログラム２１ｂが、ディスクコントローラ４ｂを介して磁気ディスク５ｂからメモリ２ｂ上にロードされる。クライアント情報処理装置１１０には、ＣＰＵ１ａ、メモリ２ａ、ＬＡＮコントローラ３ａ、ディスクコントローラ４ａ、磁気ディスク５ａ、ＩＣカードコントローラ６がそれぞれ具備されている。クライアント情報処理装置１１０の起動時には、ＯＳ２０ａ、共用ファイル暗号プログラム２１ａ、通信

26

ＡＰＩ（Ａｐｐｌｉｃａｔｉｏｎ　Ｐｒｏｇｒａｍ　Ｉｎｔｅｒｆａｃｅ）フックプログラム２７が、前記ディスクコントローラ４ａを介して前記磁気ディスク５ａから前記メモリ２ａ上にロードされる。また、ブラウザプログラム１９は、前記クライアント情報処理装置１１０のユーザー（以下単にユーザーと呼ぶ）の指示によって前記ディスクコントローラ４ａを介して前記磁気ディスク５ａから前記メモリ２ａ上にロードされる。前記通信ＡＰＩフックプログラム２７は、ブラウザプログラム１９が前記ＯＳ２０ａに発行する通信ＡＰＩをフック（横取り）し、受信ファイルデータの復号化をバックグラウンドで自動的に行うものである。

【００５８】前記サーバ情報処理装置１２０と前記クライアント情報処理装置１１０は、ローカルエリアネットワーク（ＬＡＮ）１０１を介して相互に接続されており、各々のＬＡＮコントローラ３を介して通信を行う。なお、前記サーバ情報処理装置１２０と前記クライアント情報処理装置１１０は互いに遠隔地に存在し、電話回線や専用回線を介して接続されていてもよい。

【００５９】図２８では一台のクライアント情報処理装置１１０を示したが、実際には、同じ構成のクライアント情報処理装置１１０が複数前記ＬＡＮ１０１に接続されており、これらのクライアント情報処理装置１１０間で、前記サーバ情報処理装置１２０の前記磁気ディスク５ｂに格納されたＨＴＭＬファイルやイメージデータファイルを共用する。ユーザーは前記共用を行うため、前記ブラウザプログラム１９を使用して前記磁気ディスク５ｂ上のファイルをアクセスする。なお、前記ＨＴＭＬファイルやイメージデータファイルは、前記サーバ情報処理装置１２０上で作成して前記磁気ディスク５ｂに格納しても良いし、クライアント情報処理装置１１０上で作成してから磁気ディスク５ｂに格納しても良い。また、クライアント情報処理装置１１０のＩＣカードコントローラ６は、ＩＣカード１３０が脱着可能となっている。前記ＩＣカード１３０内には、ＣＰＵ１ｃ、不揮発性メモリ７、入出力インタフェース８が具備されている。前記不揮発性メモリ７には、アクセス制御プログラム３０、グループ暗号プログラム２８、ＩＤ情報２９、マスタ鍵３１が書き込まれている。前記不揮発性メモリ７の内容は、前記入出力インタフェース８を介して外部からアクセスすることができる。ただし、不正なアクセスから前記不揮発性メモリ７の記憶内容を保護する為、前記ＣＰＵ１ｃによるユーザー認証制御が必ず介在する仕組みとなっている。

【００６０】図２９は、前記共用ファイル暗号プログラム２１と前記通信ＡＰＩフックプログラム２７を構成するプログラムルーチンを示す図である。前記共用ファイル暗号プログラム２１は、前記ＩＣカード１３０へのログイン、ログアウトを行うログイン制御ルーチン１０００、ログアウト制御ルーチン１１００、ユーザーがファ

イルの暗号化をその都度明示的に指示し暗号化する手動ファイル暗号化ルーチン２０００、前記手動ファイル暗号化をＨＴＭＬ（Ｈｙｐｅｒ　Ｔｅｘｔ　Ｍａｒｋｕｐ　Ｌａｎｇｕａｇｅ）で記述した共用ファイル向けにアレンジしたＨＴＭＬファイル暗号化ルーチン２１００から構成される。前記通信ＡＰＩフックプログラム２７は、ｓｅｎｄフックルーチン１８００、ｒｅｃｖフックルーチン１８１０、ｃｌｏｓｅ　ｓｏｃｋｅｔフックルーチン１８２０から構成される。

【００６１】前記ＩＤ情報２９の内容は前記第１の実施例における図２に示したＩＤ情報２９と同じである。ここでは、カテゴリ毎の内容を示すデータとそのコードを１０組まで登録できるようになっている。実際に前記不揮発性メモリ７に記憶するのは、データとコードの値だけである。

【００６２】次に、上記グループ暗号プログラム２８の動作は前記実施例におけるグループ暗号プログラム２８の動作と、宛先リストのハッシュ値を計算し前記グループ鍵５１を作成する手順は前記実施例における図３に示した手順とそれぞれ同じである。前記ＩＣカード１３０内の前記グループ暗号プログラム２８の処理内容は前記実施例における図４に示したフロー図と同じである。以上説明したグループ鍵５１を用いてデータを暗号化することによって、前記宛先リスト４１に含まれるＩＤ情報２９を持つ者だけが該データの復号化を可能とすることができる。

【００６３】次に、上述のグループ暗号をファイルの暗号に適用した例について説明する。図３０は、ファイルの暗号化手順と暗号化後のファイルの構成を示す図である。図５に示した例とは、暗号化後のファイル５０２のヘッダ５０５の内容が相違している。すなわち、５０１は暗号化する前の平文ファイル、５０２は暗号化後の暗号ファイルである。６１は乱数として生成された実効鍵、５０４は前記実効鍵６１を鍵として前記ファイル５０１の内容を暗号化した暗号文データ、５０５は暗号文データ５０４に付加するヘッダ、５０６から５１１は前記ヘッダ５０５の内容であり、５０６は使用した暗号化方式を示す文字列、５０９は前記実効鍵６１をグループ鍵５１で暗号化した暗号化実効鍵、５１１は前記宛先リスト４１を前記マスタ鍵３１で暗号化した暗号化宛先リスト、５１０は前記暗号化宛先リスト５１１のサイズである。前記暗号ファイル５０２は、図３１に示すように上記手順と全く逆の手順で復号化することができる。

【００６４】上述のファイル暗号化においては、前記グループ鍵５１は前記実効鍵６１を暗号化する為に用い、ファイルデータの暗号化には前記実効鍵６１を用いた。したがって、前記宛先リスト４１を変更する場合には、まず前記実効鍵６１を変更前の前記宛先リスト４１で生成した前記グループ鍵５１で復号化し、次に変更後の前記宛先リスト４１から生成した前記グループ鍵５１で再暗号化すればよい。これにより、前記宛先リスト４１の変更時にも前記実効鍵６１に比べデータ量の多いファイルデータは再暗号化が不要とすることができ、変更処理を高速に行うことができる。

【００６５】次に、共用ファイル暗号プログラム２１内のログイン処理ルーチン１０００、ログアウト処理ルーチン１１００について説明する。前記ＩＣカード１３０へのログイン処理ルーチン１０００の処理フローは、図１０に示した第１の実施例におけるログイン処理ルーチン１０００と同じであり、詳細な説明は省略する。タイマイベントを用いて定期的に前記ＩＣカード１３０の接続を調べることによって前記ＩＣカード１３０を抜いた際のログアウト処理を自動化することができる。

【００６６】前記ＩＣカード１３０からのログアウト処理ルーチン１１００の処理フローは、図１１に示した第１の実施例におけるログアウト処理ルーチン１１００と同じであり、詳細な説明は省略する。

【００６７】図３２は、本実施例のファイル暗号化・復号化におけるデータの入出力方法を示す図である。前記暗号ファイル５０２の前記暗号文データ５０４は、暗号文ブロックバッファ８２のサイズ単位で前記暗号文ブロックバッファ８２との間で読み出し・書き込みが行われる。前記暗号文ブロックバッファ８２のデータは復号化し平文ブロックバッファ８１に転送される。また、前記平文ブロックバッファ８１のデータは暗号化し前記暗号文ブロックバッファ８２に転送される。この入出力方法が図１８に示した入出力方法と異なる点は、バッファ長が１０２４ｂｙｔｅから２５６ｂｙｔｅとして示されている点である。

【００６８】次に、前記共用ファイル暗号プログラム２１に含まれる手動ファイル暗号化ルーチン２０００について説明する。このルーチンは、暗号化を行う平文ファイルのファイル名称と、暗号化後の暗号ファイル名称と宛先リスト４１を入力として動作する。この処理は、図２０に示したファイル暗号化ルーチン２０００と略同じであり、詳細な説明を省略する。

【００６９】次に、ＨＴＭＬファイルの暗号化を行う実施例について説明する。この実施例においても、平文のＨＴＭＬファイルおよびブラウザでの表示ならびに暗号化したＨＴＭＬファイルなどは、実施例１の図２２～図２６に示した例と同じであり、その詳細な説明を省略する。

【００７０】上記のＨＴＭＬファイル暗号化を行うＨＴＭＬファイル暗号化プログラムの処理フローは実施例１の図２１に示した処理フローと同じであり、詳細な説明を省略する。この方法で作成した暗号化ＨＴＭＬファイルは図２６に示したものと同じである。以上の暗号化処理を施したファイルは、情報提供者によって前記サーバ情報処理装置の磁気ディスク５ｂに格納される。

【００７１】次に、前記ブラウザプログラム１９が、Ｗ

29

WWWサーバプログラム２６から前記暗号ファイルを受信する際の通信ＡＰＩフックプログラム２７の処理について説明する。まず、受信ファイルデータの入出力とその管理方法から説明する。図３３は、本実施例における受信ファイル復号化の際のデータ入出力を示す図である。８３は、ブラウザプログラム１９が用意する暗号文受信バッファであり、暗号文が含まれている。通信ＡＰＩフックモジュールは、後述するｒｅｃｖフック処理の中で、受信した前記暗号文を暗号文ブロックバッファ８２にコピーする。このとき、暗号文ブロックバッファ８２が満たされた場合、該暗号文の復号化処理を実行し平文を平文ブロックバッファ８１にコピーする。さらに、平文をブラウザ２３へ渡すために、前記平文ブロックバッファ８１の中身を前記暗号文受信バッファに上書きコピーする。暗号文受信バッファ８３から暗号文ブロックバッファ８２へコピーした暗号文が、前記暗号文ブロックバッファ８２を満たしていない場合は、復号化処理を行わずに次の受信ファイルデータを待つ。以後、受信した暗号ファイルデータの前記暗号文ブロックバッファ８２へのコピー、前記復号化処理と平文ブロックバッファ８１へのコピー、および暗号文受信バッファ８３への平文コピーを、受信した暗号ファイルデータがなくなるまで繰り返す。

【００７２】図３４は、オープン中のソケットを管理するためのソケット情報テーブル９１である。ソケット情報テーブル９１は、ソケット記述子１９１２、受信フラグ１９１３、受信データポインタ１９１４、暗号化実効鍵５０９、暗号化宛先リストのサイズ５１０、暗号化宛先リスト５１１、グループ鍵５１、実効鍵６１、暗号文ブロックバッファポインタ１９１５、平文ブロックバッファポインタ１９１６の各情報を、オープン中の各ソケット毎にポインタ１９０１を用いてリスト構造で記憶している。ここでソケット記述子１９１２は、ソケットをオープンした際に前記ＯＳ２０から与えられる管理番号である。受信フラグ１９１３とは、前記ソケット記述子１９１２を使用して暗号ファイルのデータを１度も受信していない場合には”０”、暗号ファイルのデータを１回以上受信した場合には”１”とするフラグである。前記受信データポインタ１９１４は、受信したファイルデータから暗号文を読み込む際の起点を示すものであり、後述するｒｅｃｖフックルーチン１８１０が、データ受信時に受信バッファの先頭アドレスを登録する。暗号文ブロックバッファポインタ１９１５と平文ブロックバッファポインタ１９１６は、それぞれ図３３における暗号文ブロックバッファ８２と平文ブロックバッファ８１へのポインタであり、復号化処理時にバッファ間で暗号文や平文をコピーする際の起点となるものである。また、前記暗号化実効鍵５０９と前記暗号化宛先リストサイズ５１０、および暗号化宛先リスト５１１は、暗号ファイル受信時に前記ヘッダー５０５から読み込んで登録する。なお、該暗号化宛先リスト５１１は、図６に示すようにマスタ鍵によって復号化される。更に、該宛先リスト４１からグループ鍵５１を生成し、また前記暗号化実効鍵５０９を前記グループ鍵５１で復号化して実効鍵６１を求め、それぞれソケット情報テーブル９１に登録する。

30

【００７３】次に、前記通信ＡＰＩフックプログラム２７内の各ルーチンの処理について説明する。図３５は、前記ｓｅｎｄフックルーチン１８００の処理フローを示す図である。まず、ステップ１８０１で、ブラウザプログラム１９がWWWサーバプログラム２６へ送信するデータ中にファイル取得用コマンドである”ＧＥＴ”が存在するか否かを調べる。存在すればステップ１８０４で前記ソケット情報テーブル９１を参照し、フックしたソケットの記述子が登録されているか否かを調べる。登録されていれば、ステップ１８０６で該当するソケット情報のうち、受信フラグ１９１３を”０”にして、ステップ１８０７のｓｅｎｄ処理を行う。登録されていない場合、ステップ１８０５でソケット情報をテーブル９０に登録する。前記ステップ１８０１で、送信データ中に前記ファイル取得用コマンドが存在しなければ、フックしたソケットの記述子が前記ソケット情報テーブルに登録されているか否かをステップ１８０２で調べる。登録されていれば、ステップ１８０３で該当するソケット情報を削除する。登録されていなければ、ステップ１８０７でｓｅｎｄ処理を実行する。次に、ステップ１８０８でｓｅｎｄ処理のエラーが発生し、かつステップ１８０９で該当するソケット情報がテーブルに登録済みの場合は、ステップ１８１０で該当するソケット情報を前記ソケット情報テーブル９１から削除し、ステップ１８１１で前記ブラウザプログラム１９にリターンする。前記ステップ１８０８でエラーが発生しなければ、なにもせずにステップ１８１１で前記ブラウザプログラム１９にリターンする。

【００７４】図３６は、ｒｅｃｖフックルーチン１８２０の処理フローを示す図である。まず、ステップ１８２１でブラウザプログラム１９がコールしたｒｅｃｖの処理を行い、ステップ１８２２で前記ｒｅｃｖ処理のエラーチェックを行う。このときエラーが発生した場合、ステップ１８３５で前記エラーを前記ブラウザプログラム１９にリターンする。なお、エラーに関するメッセージは前記ブラウザプログラム１９が表示するので、ｒｅｃｖフックルーチン１８１０がエラーメッセージを表示する必要はない。エラーなしの場合は、ステップ１８２３でソケットの記述子が前記ソケット情報テーブル９１に登録済みであるかどうかを調べる。ここで、ｓｅｎｄフックルーチンの中でソケット情報テーブル９１に登録されたソケット記述子は、WWWサーバ２６が管理するＨＴＭＬファイルやイメージデータファイルの受信に使用するソケットを指す。そこで、ソケット情報テーブル９

１に登録済みの場合はステップ１８２４で受信データの有無をチェックする。登録されてない場合は、ステップ１８３５で受信データをブラウザプログラム１９にリターンする。ステップ１８２４で受信データが存在しない場合は、データの受信が完了したことを意味するので、該当するソケット情報を前記ソケット情報テーブル９１から削除し、ステップ１８３５で前記ブラウザプログラム１９にリターンする。受信データが存在する場合は、ステップ１８２５でｒｅｃｖフラグをチェックする。前記ｒｅｃｖフラグが”０”であれば、前記ｓｅｎｄ処理に対する１回目の受信データであると判断し、ステップ１８２６で前記受信データをチェックする。ステップ１８２７で、受信データが本発明の暗号化方式により暗号化されたファイルである場合、ステップ１８２８でログイン中のユーザーが存在するかチェックする。存在しない場合は、ステップ１８３８でメッセージ１８３７を表示する。存在する場合は、ステップ１８２９で宛先リストに、前記ログインユーザーが含まれているかを調べる。前記ログインユーザーが含まれている場合は、ステップ１８３０で前記ｒｅｃｖフラグを”１”に設定し、ステップ１８３２で前記受信ファイルデータを復号化する。一方、前記宛先リストに前記ログインユーザーが含まれていない場合は、ステップ１８３１でメッセージ１８３６を表示し、ステップ１８３４でソケット情報を前記ソケット情報テーブル９１から削除してリターンする。一方、前記ステップ１８２７で、受信ファイルデータが暗号化されていない場合は、ステップ１８３４でソケット情報を前記ソケット情報テーブルから削除してリターンする。

【００７５】図３７は、ｃｌｏｓｅｓｏｃｋｅｔフックルーチン１８４０の処理フローを示す図である。まずステップ１８４１では、前記ブラウザプログラム１９がコールしたｃｌｏｓｅｓｏｃｋｅｔ関数の引数に指定されたソケット記述子が、ソケット情報テーブル９１に登録されているかを調べる。登録されている場合は、ステップ１８４２で前記ソケット記述子に対応するソケット情報をテーブル９０から削除し、ステップ１８４３のｃｌｏｓｅｓｏｃｋｅｔ処理を実行して前記ブラウザプログラム１９にリターンする（ステップ１８４４）。一方、前記ソケット情報テーブルに登録されていない場合は、ステップ１８４３のｃｌｏｓｅｓｏｃｋｅｔ処理を実行してリターンする。

【００７６】以上説明した通信ＡＰＩフックプログラム２７の各ルーチンの処理によって、所望のページの復号化をブラウザプログラム１９からは透過に実現することができる。

【００７７】次に、本実施例において、実際のWWW運用システムの一例について図３８を用いて説明する。１４０は、代理サーバ情報処理装置であり、代理サーバプログラム１８により、前記クライアント情報処理装置１１０と前記サーバ情報処理装置１２０との間の送受信データを中継する。また、前記代理サーバプログラム１８は、前記サーバ情報処理装置１２０が前記クライアント情報処理装置１１０からの要求に応じて発信したＨＴＭＬファイルやイメージデータファイルを、ディスクコントローラ４ｃを介して磁気ディスク５ｃ内にキャッシュする機能を持つ。以後、前記クライアント情報処理装置１１０からファイル転送要求を出した場合、該ファイルが前記磁気ディスク５ｃ内にキャッシュされていれば、前記代理サーバプログラム１８が該キャッシュファイルを前記クライアント情報処理装置１１０へ転送する。本実施例においては、機密情報に関するファイルは全て暗号化されて前記サーバ情報処理装置１２０の磁気ディスク５ｂに格納されている。したがって、ファイル転送時に前記磁気ディスク５ｃにキャッシュされる機密情報は全て暗号ファイルとなる。

【００７８】

【発明の効果】以上説明したように、本発明はファイルの暗号化において、復号化を許可するユーザーのリストをファイルに付加するグループ暗号用いるので、複数のユーザーが同一のファイル利用する場合にも暗号化ファイルが一個で済み且つ安全であるという効果がある。

【００７９】また、暗号化の対象ディレクトリと復号を許可するユーザーの宛先リストをユーザー毎に設定可能であるので、複数のユーザーがディレクトリを共用し該ディレクトリ内のファイルを異なる宛先リストを用いてファイル暗号化を行うことができるという効果がある。

【００８０】また、本発明は、アプリケーションプログラムが解釈し処理する文法に従って記述された平文ファイルのデータ暗号化において、前記文法における前記アプリケーションプログラム処理時に処理対象外として無視する注釈文とするための記述子を前記暗号化後のデータに付加し暗号ファイルとして格納するので、本暗号化システムに対応した復号プログラムを具備しない情報処理装置で前記暗号化ファイルを読み出した場合にも、異常動作することがないという効果がある。

【００８１】また、本発明は、アプリケーションプログラムからＯＳに発行されるファイル操作要求をフックするフック手段を設け、前記ファイル操作要求発行時に自動的に暗号化あるいは復号化を行うので、ユーザーが暗号化を忘れることがないという効果がある。

【００８２】また、本発明は、WWWサーバが予め暗号化されたファイルを管理するので、該WWWサーバプログラムを具備するサーバ情報処理装置に直接アクセスしても機密情報が漏れないという効果がある。また、本発明は、WWWサーバが予め暗号化されたファイルを管理するので、該ファイル転送時に代理サーバによってキャッシュされる機密情報も暗号ファイルとなるため、代理サーバ情報処理装置に直接アクセスしても機密情報が漏れないという効果がある。

【００８３】また、本発明は、ブラウザプログラムから通信モジュールに発行される通信ＡＰＩ呼び出しをフックするフック手段を設け、前記データ受信ＡＰＩ呼び出し時に自動的に復号化を行うので、ユーザーにとって復号化のための操作が不要であるという効果がある。

【図面の簡単な説明】

【図１】本発明の実施例における共用ファイル暗号システムの一構成例を示す図。

【図２】本発明の実施例におけるＩＤ情報２９の内容の一例を示す図。

【図３】本発明の実施例において、宛先リストのハッシュ値を計算し前記グループ鍵５１を作成する手順の一例を示す図。

【図４】本発明の一実施例におけるＩＣカード１３０内の前記グループ暗号プログラム２８の処理内容を示すフロー図。

【図５】本発明の一実施例におけるファイルの暗号化手順と暗号化後のファイルの構成を示す図。

【図６】本発明の一実施例におけるファイルの復号化手順を示す図。

【図７】本発明の一実施例におけるユーザーから見た論理的なディレクトリの構成を示す図。

【図８】本発明の一実施例におけるファイルの自動暗号化、復号化を導入した後の物理的なディレクトリ構成を示す図。

【図９】本発明の一実施例における宛先リストファイル９０１、前記暗号ファイル情報ファイル９０２の内容の一例を示す図。

【図１０】本発明の一実施例におけるＩＣカード１３０へのログイン処理ルーチン１０００の処理フローを示す図。

【図１１】本発明の一実施例におけるＩＣカード１３０からのログアウト処理ルーチン１１００の処理フローを示す図。

【図１２】本発明の一実施例におけるファイルオープンフックルーチン１２００の処理フローを示す図。

【図１３】本発明の一実施例におけるファイルクリエイトフックルーチン１３００の処理フローを示す図。

【図１４】本発明の一実施例におけるリードフックルーチン１４００の処理フローを示す図。

【図１５】本発明の一実施例におけるライトフックルーチン１５００の処理フローを示す図。

【図１６】本発明の一実施例におけるクローズフックルーチン１６００の処理フローを示す図。

【図１７】本発明の一実施例におけるファイルリネームフックルーチン１７００の処理フローを示す図。

【図１８】本発明の一実施例におけるファイル暗号化・復号化時のデータの入出力方法を示す図。

【図１９】本発明の一実施例におけるオープン中の暗号ファイルを管理するためのファイルハンドルテーブル９０の構成を示す図。

【図２０】本発明の一実施例における手動ファイル暗号化ルーチン２０００の処理フローの一例を示す図。

【図２１】本発明の第一の実施例における第一のサーバ情報処理装置１２０のディレクトリ構造を示す図。

【図２２】本発明の一実施例における平文のＨＴＭＬファイルの一例を示す図。

【図２３】本発明の一実施例において平文のＨＴＭＬファイルをブラウザで表示した画面の例を示す図。

【図２４】本発明の一実施例における暗号化したＨＴＭＬファイルの一例を示す図。

【図２５】本発明の一実施例において暗号化した平文のＨＴＭＬファイルをブラウザで表示した画面の例を示す図。

【図２６】本発明の一実施例における暗号化したＨＴＭＬファイルの他の例を示す図。

【図２７】本発明の第一の実施例における前記共用ファイル暗号プログラム２１と前記ファイルＩ／Ｏフックプログラム２２を構成するプログラムルーチンを示す図。

【図２８】本発明の実施例における暗号ファイル受信システムの一構成例を示す図。

【図２９】本発明の実施例における前記共用ファイル暗号プログラム２１と前記通信ＡＰＩフックプログラム２７を構成するプログラムルーチンを示す図。

【図３０】本発明の一実施例におけるファイルの暗号化手順と暗号化後のファイルの構成を示す図。

【図３１】本発明の一実施例におけるファイルの復号化手順を示す図。

【図３２】本発明の一実施例におけるファイル暗号化・復号化時のデータの入出力方法を示す図。

【図３３】本発明の一実施例における受信ファイル復号化時のデータの入出力方法を示す図。

【図３４】本発明の一実施例におけるオープン中のソケット情報を管理するためのソケット情報テーブル９１の構成を示す図。

【図３５】本発明の一実施例におけるｓｅｎｄフックルーチン１８００の処理フローを示す図。

【図３６】本発明の一実施例におけるｒｅｃｖフックルーチン１８２０の処理フローを示す図。

【図３７】本発明の一実施例におけるｃｌｏｓｅｓｏｃｋｅｔフックルーチン１８４０の処理フローを示す図。

【図３８】本発明の実施例において実際のWWWを運用するシステムの一構成例を示す図。

【符号の説明】

１　ＣＰＵ
２　メモリ
３　ＬＡＮコントローラ
４　ディスクコントローラ
５　磁気ディスク
６　ＩＣカードコントローラ

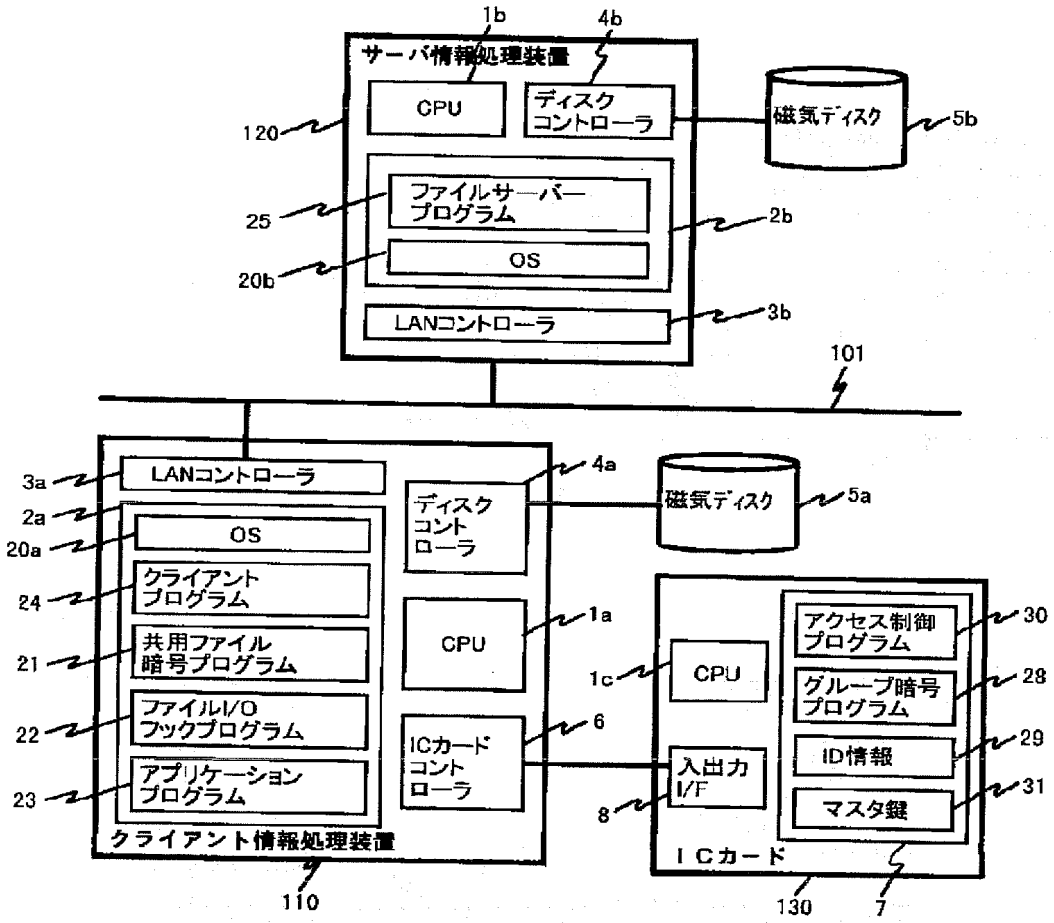
35

７　不揮発メモリ
８　入出力インタフェース
１９　ブラウザプログラム
１８　代理サーバプログラム
２０　ＯＳ
２１　共用ファイル暗号プログラム
２２　ファイルＩ／Ｏフックプログラム
２３　アプリケーションプログラム
２４　クライアントプログラム
２５　ファイルサーバプログラム
２６　WWWサーバプログラム

36

２７　通信ＡＰＩフックプログラム
２８　グループ暗号プログラム
２９　ＩＤ情報
３０　アクセス制御プログラム
３１　マスタ鍵
３２　ハッシュ関数
１０１　ＬＡＮ
１１０　クライアント情報処理装置
１２０　サーバ情報処理装置
１３０　ＩＣカード
１４０　代理サーバ情報処理装置

【図１】

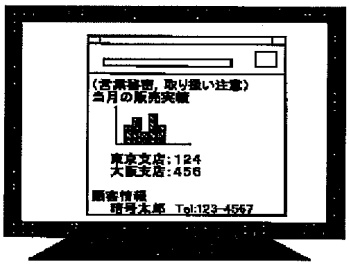


【図２】

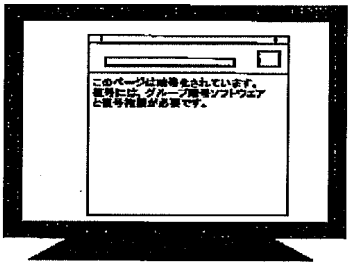
ＩＤ情報　29

	カテゴリ	データ	コード
1	生年月日	19900101	–
2	氏　名	復号次郎	–
3	性　別	男性	1
4	氏名コード	677991234	–
5	事業所	システム研究所	676
6	部	第四部	04
7	課	安全課	402
8	職名	主任	5
9	分野	セキュリティ	X5
10	出身地	神奈川	24

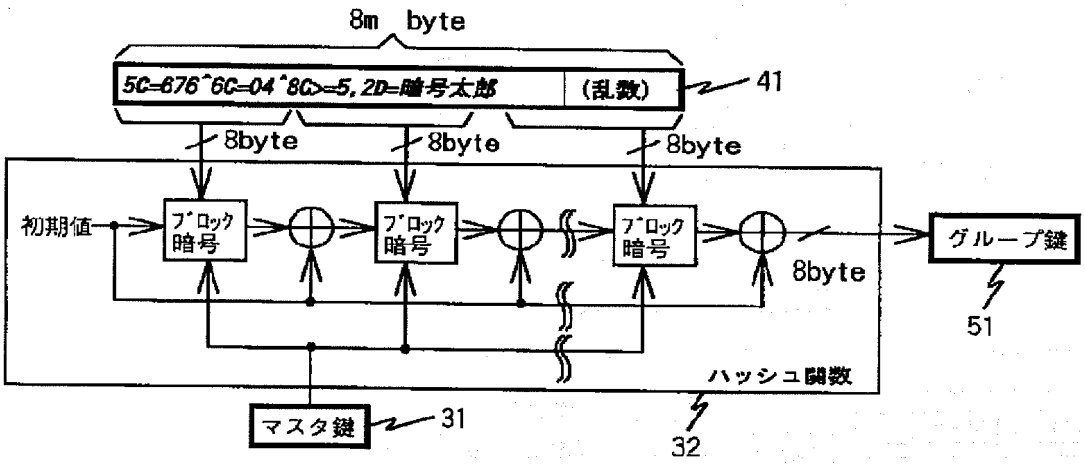
【図２３】



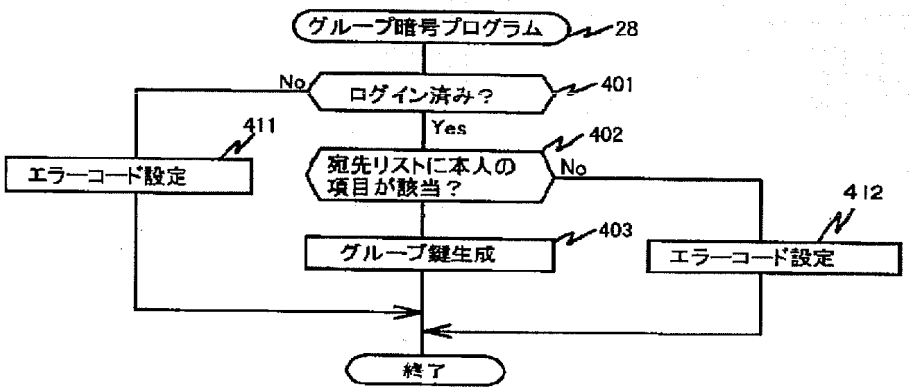
【図２５】



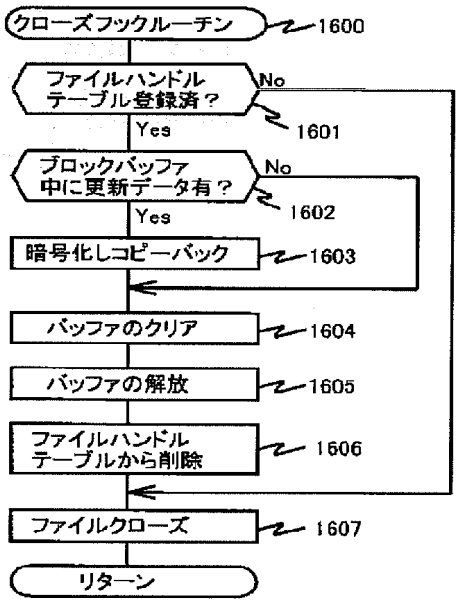
【図３】



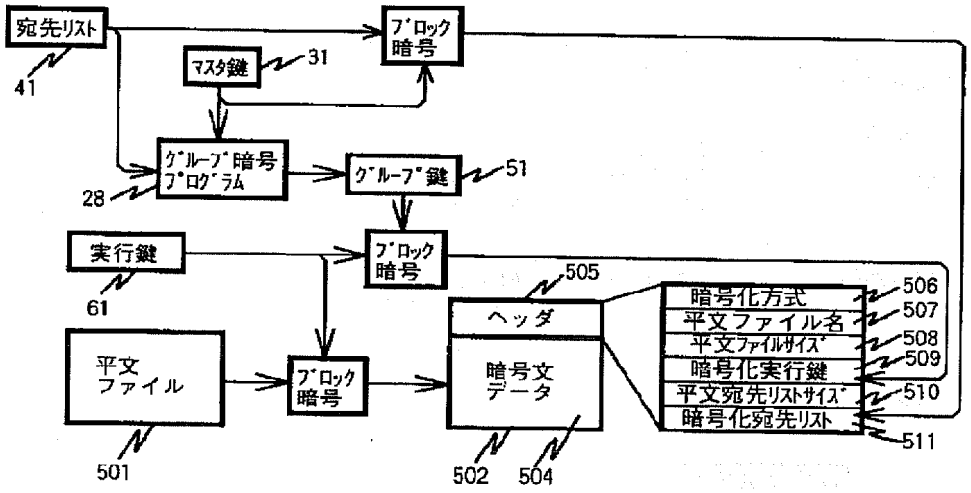
【図４】



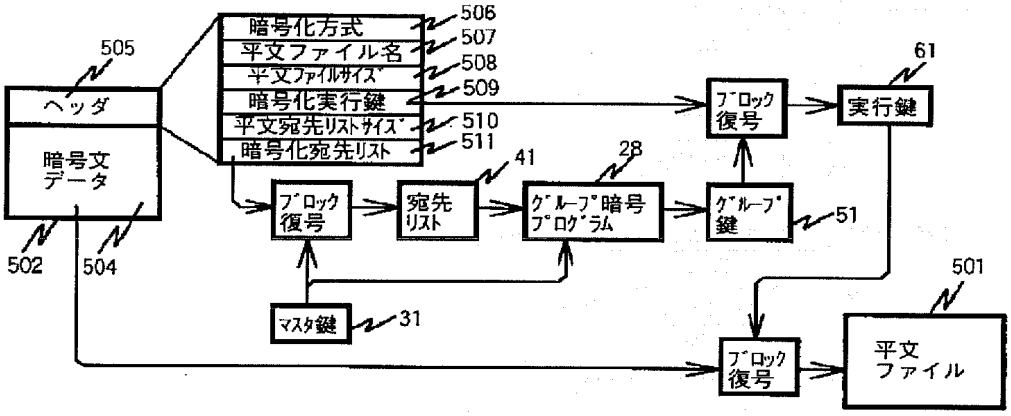
【図１６】



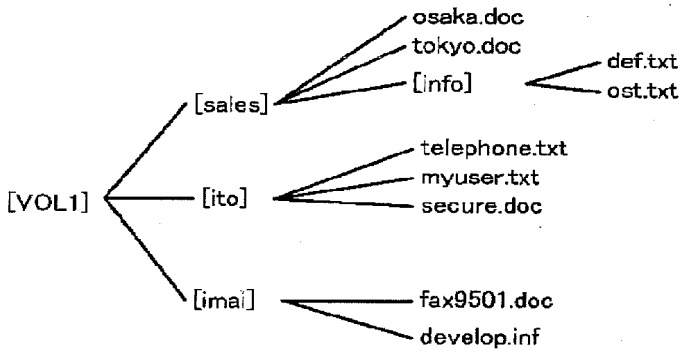
【図５】


宛先リスト
41
ﾌﾞﾛｯｸ暗号
マスタ鍵
31
ｸﾞﾙｰﾌﾟ暗号ﾌﾟﾛｸﾞﾗﾑ
28
ｸﾞﾙｰﾌﾟ鍵
51
実行鍵
61
ﾌﾞﾛｯｸ暗号
505
ヘッダ
暗号化方式
平文ファイル名
平文ﾌｧｲﾙｻｲｽﾞ
暗号化実行鍵
平文宛先ﾘｽﾄｻｲｽﾞ
暗号化宛先リスト
506
507
508
509
510
511
平文ファイル
501
ﾌﾞﾛｯｸ暗号
暗号文データ
502
504

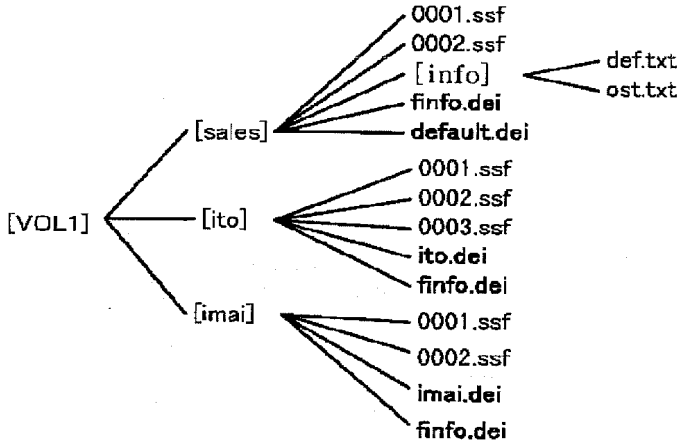
【図６】


505
ヘッダ
暗号文データ
502
504
暗号化方式
平文ファイル名
平文ﾌｧｲﾙｻｲｽﾞ
暗号化実行鍵
平文宛先ﾘｽﾄｻｲｽﾞ
暗号化宛先リスト
506
507
508
509
510
511
ﾌﾞﾛｯｸ復号
61
実行鍵
41
28
ﾌﾞﾛｯｸ復号
宛先リスト
ｸﾞﾙｰﾌﾟ暗号ﾌﾟﾛｸﾞﾗﾑ
ｸﾞﾙｰﾌﾟ鍵
51
マスタ鍵
31
501
ﾌﾞﾛｯｸ復号
平文ファイル

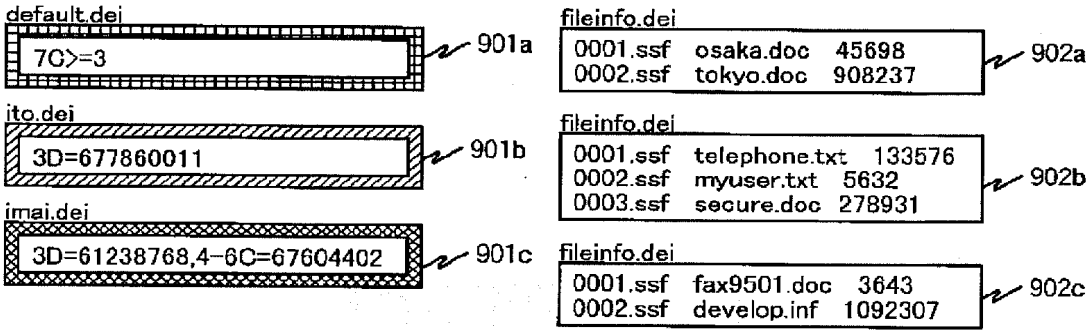
【図７】


[VOL1]
[sales]
osaka.doc
tokyo.doc
[info]
def.txt
ost.txt
[ito]
telephone.txt
myuser.txt
secure.doc
[imai]
fax9501.doc
develop.inf

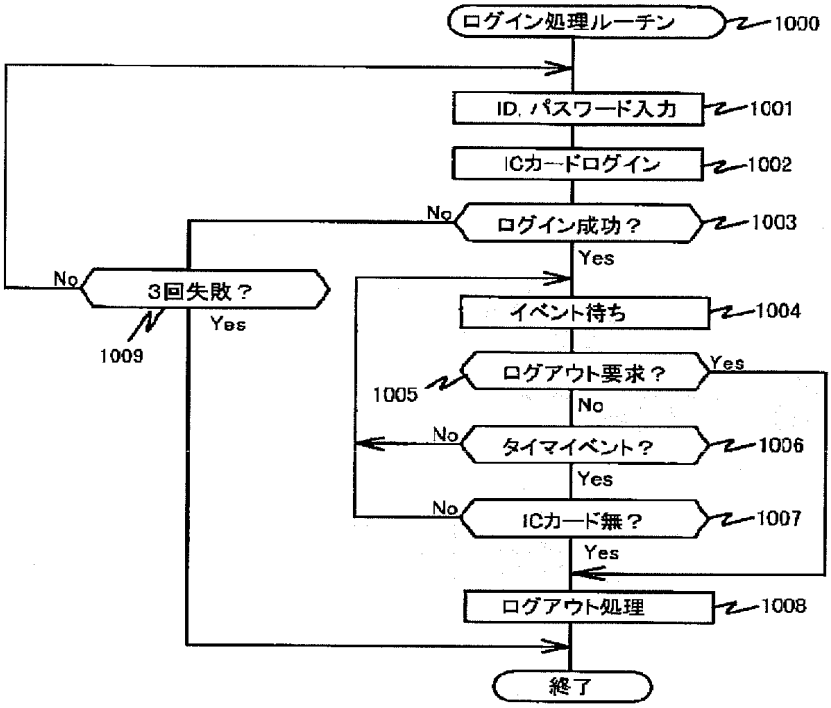
【図８】


[VOL1]
[sales]
0001.ssf
0002.ssf
[info]
def.txt
ost.txt
finfo.dei
default.dei
[ito]
0001.ssf
0002.ssf
0003.ssf
ito.dei
finfo.dei
[imai]
0001.ssf
0002.ssf
imai.dei
finfo.dei

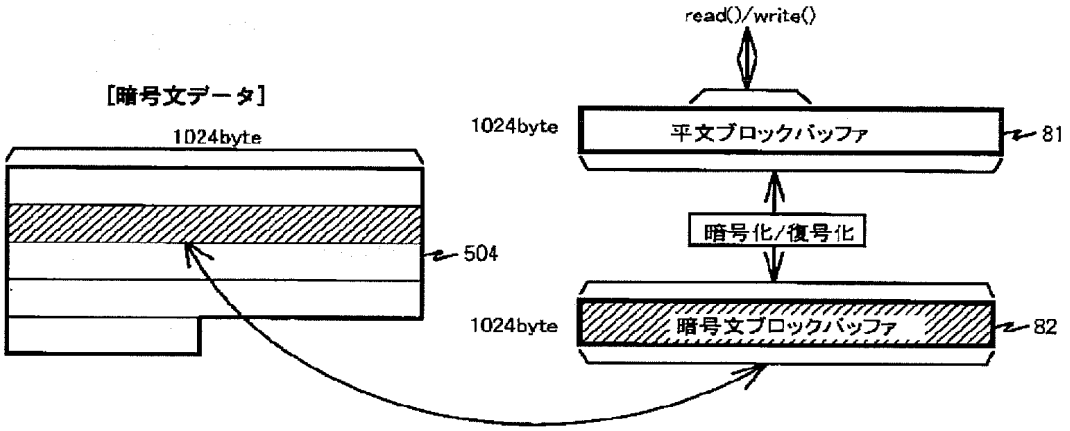
【図９】


default.dei
7C>=3
901a
ito.dei
3D=677860011
901b
imai.dei
3D=61238768,4-6C=67604402
901c
fileinfo.dei
0001.ssf osaka.doc 45698
0002.ssf tokyo.doc 908237
902a
fileinfo.dei
0001.ssf telephone.txt 133576
0002.ssf myuser.txt 5632
0003.ssf secure.doc 278931
902b
fileinfo.dei
0001.ssf fax9501.doc 3643
0002.ssf develop.inf 1092307
902c

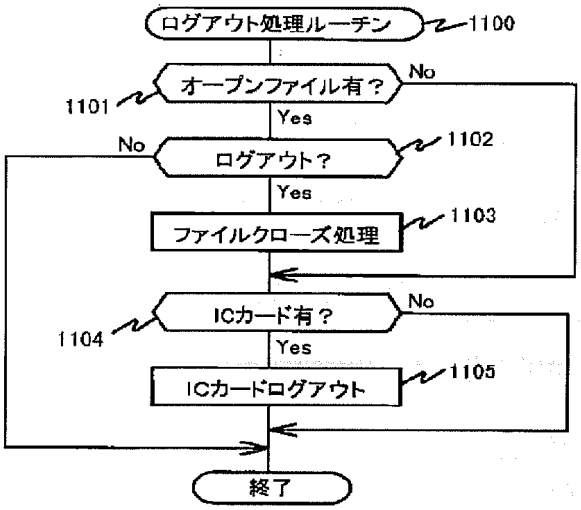
【図１０】


ログイン処理ルーチン
1000
ID, パスワード入力
1001
ICカードログイン
1002
No
ログイン成功？
1003
Yes
No
3回失敗？
Yes
1009
イベント待ち
1004
1005
ログアウト要求？
Yes
No
No
タイマイベント？
1006
Yes
No
ICカード無？
1007
Yes
ログアウト処理
1008
終了

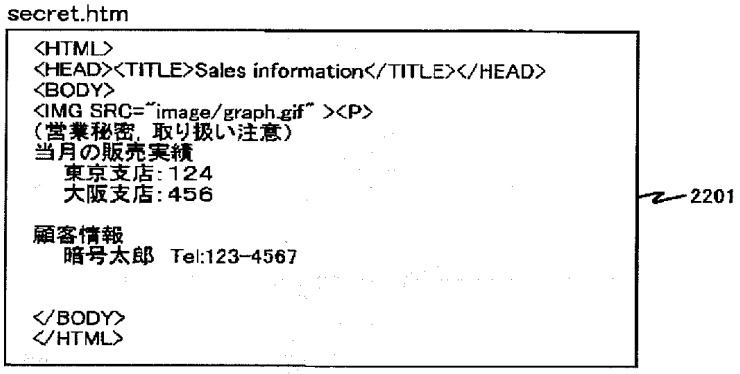
【図１８】


read()/write()
[暗号文データ]
1024byte
504
1024byte
平文ブロックバッファ
81
暗号化/復号化
1024byte
暗号文ブロックバッファ
82

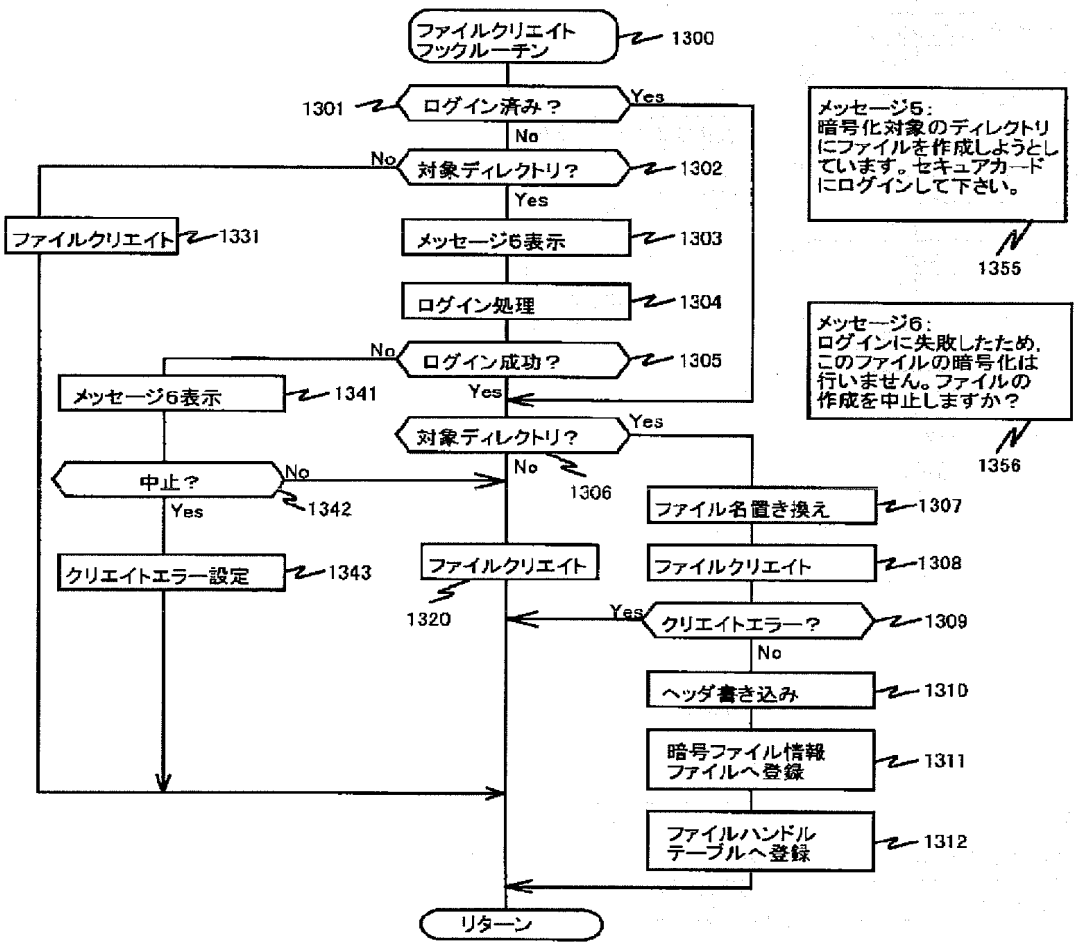
【図１１】


ログアウト処理ルーチン
1100
オープンファイル有？
No
1101
Yes
No
ログアウト？
1102
Yes
ファイルクローズ処理
1103
ICカード有？
No
1104
Yes
ICカードログアウト
1105
終了

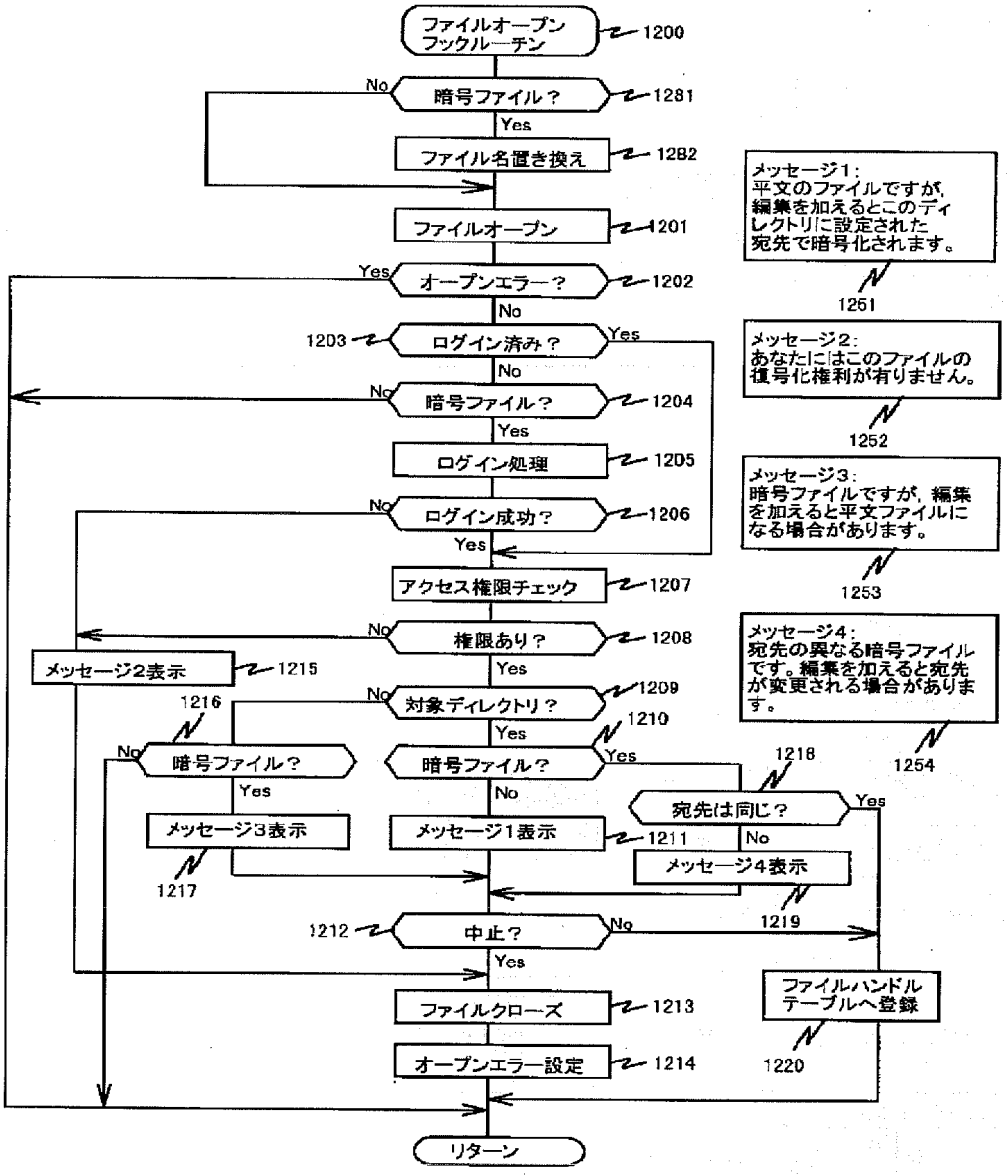
【図２２】


secret.htm
<HTML>
<HEAD><TITLE>Sales information</TITLE></HEAD>
<BODY>
<IMG SRC="image/graph.gif" ><P>
(営業秘密，取り扱い注意)
当月の販売実績
東京支店：124
大阪支店：456
顧客情報
暗号太郎　Tel:123-4567
</BODY>
</HTML>
2201

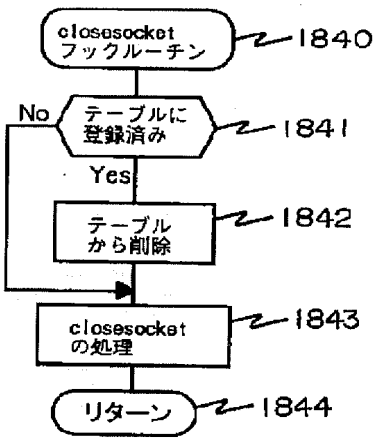
【図１３】


ファイルクリエイト
フックルーチン
1300
1301
ログイン済み？
Yes
No
No
対象ディレクトリ？
1302
Yes
ファイルクリエイト
1331
メッセージ5表示
1303
ログイン処理
1304
No
ログイン成功？
1305
メッセージ6表示
1341
Yes
対象ディレクトリ？
Yes
No
1306
中止？
No
1342
Yes
ファイル名置き換え
1307
クリエイトエラー設定
1343
ファイルクリエイト
1320
ファイルクリエイト
1308
Yes
クリエイトエラー？
1309
No
ヘッダ書き込み
1310
暗号ファイル情報
ファイルへ登録
1311
ファイルハンドル
テーブルへ登録
1312
リターン
メッセージ5：
暗号化対象のディレクトリにファイルを作成しようとしています。セキュアカードにログインして下さい。
1355
メッセージ6：
ログインに失敗したため，このファイルの暗号化は行いません。ファイルの作成を中止しますか？
1356

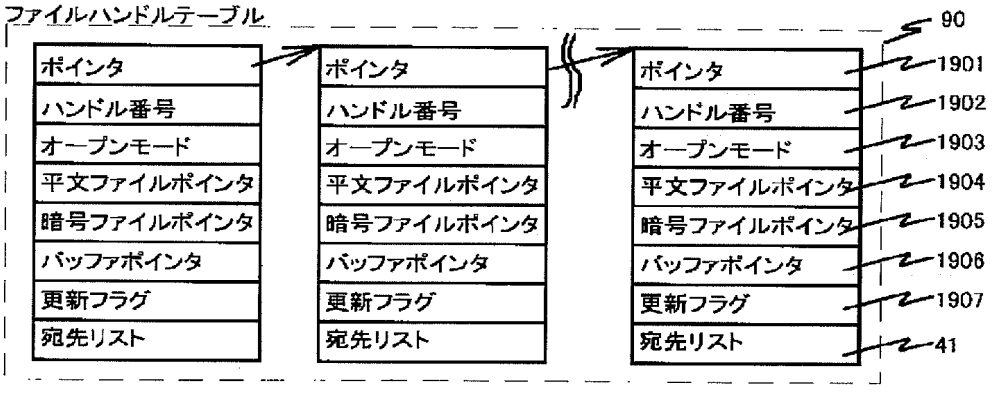
【図１２】


ファイルオープン フックルーチン
1200
暗号ファイル？
1281
No
Yes
ファイル名置き換え
1282
ファイルオープン
1201
オープンエラー？
1202
Yes
No
1203
ログイン済み？
Yes
No
暗号ファイル？
1204
Yes
ログイン処理
1205
ログイン成功？
1206
No
Yes
アクセス権限チェック
1207
権限あり？
1208
No
Yes
メッセージ2表示
1215
対象ディレクトリ？
1209
No
Yes
1216
1210
暗号ファイル？
No
Yes
暗号ファイル？
Yes
No
メッセージ3表示
1217
メッセージ1表示
1211
1218
宛先は同じ？
Yes
No
メッセージ4表示
1219
1212
中止？
No
Yes
ファイルクローズ
1213
オープンエラー設定
1214
ファイルハンドル テーブルへ登録
1220
リターン
メッセージ1: 平文のファイルですが, 編集を加えるとこのディレクトリに設定された宛先で暗号化されます。
1251
メッセージ2: あなたにはこのファイルの復号化権利が有りません。
1252
メッセージ3: 暗号ファイルですが, 編集を加えると平文ファイルになる場合があります。
1253
メッセージ4: 宛先の異なる暗号ファイルです。編集を加えると宛先が変更される場合があります。
1254

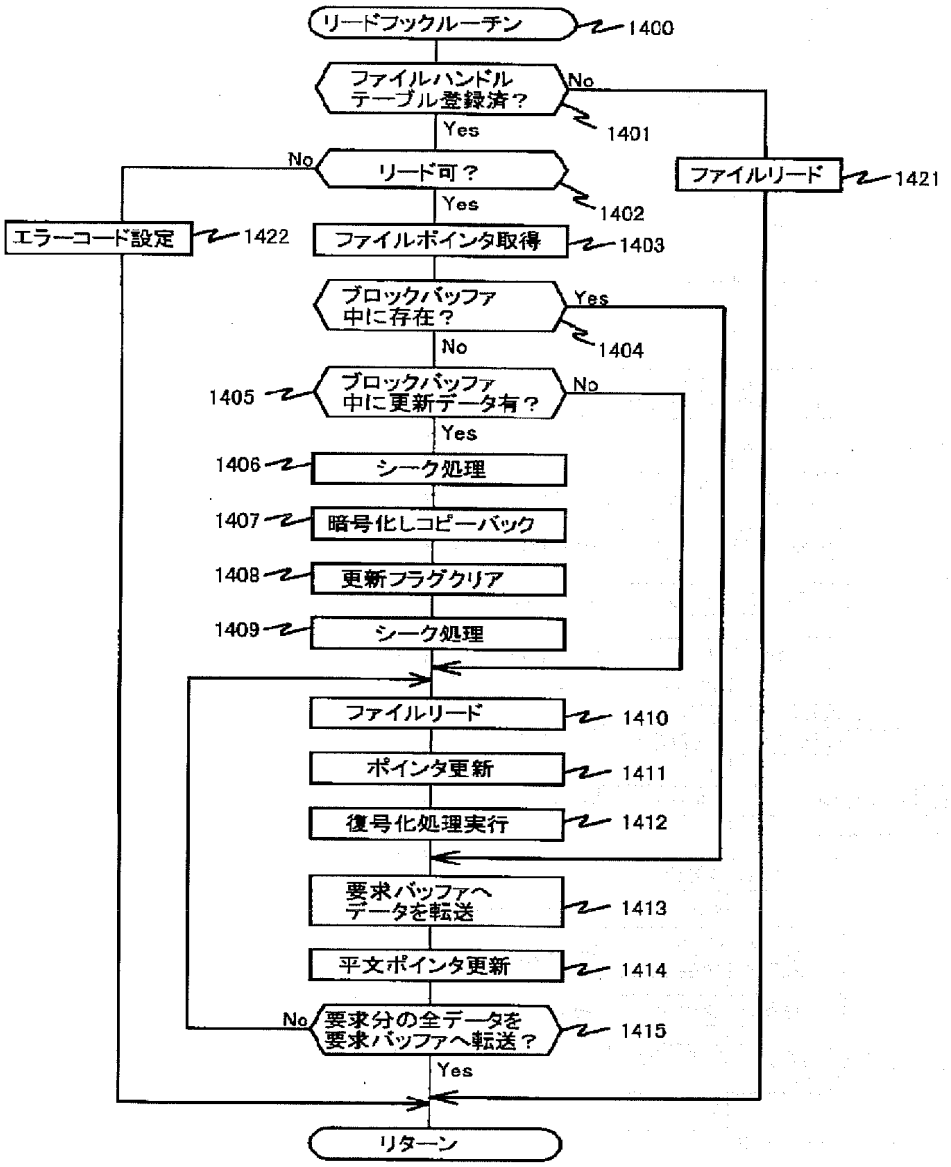
【図３７】


closesocket フックルーチン
1840
No
テーブルに 登録済み
1841
Yes
テーブル から削除
1842
closesocket の処理
1843
リターン
1844

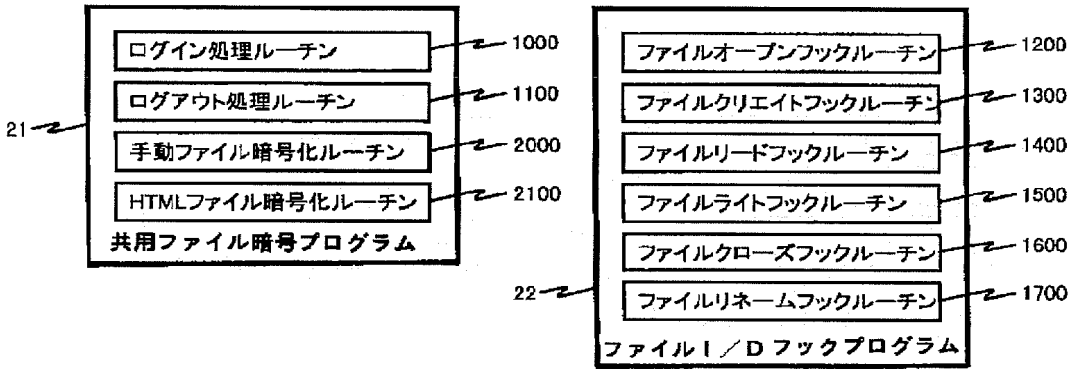
【図１９】


ファイルハンドルテーブル
90
ポインタ
ハンドル番号
オープンモード
平文ファイルポインタ
暗号ファイルポインタ
バッファポインタ
更新フラグ
宛先リスト
1901
1902
1903
1904
1905
1906
1907
41

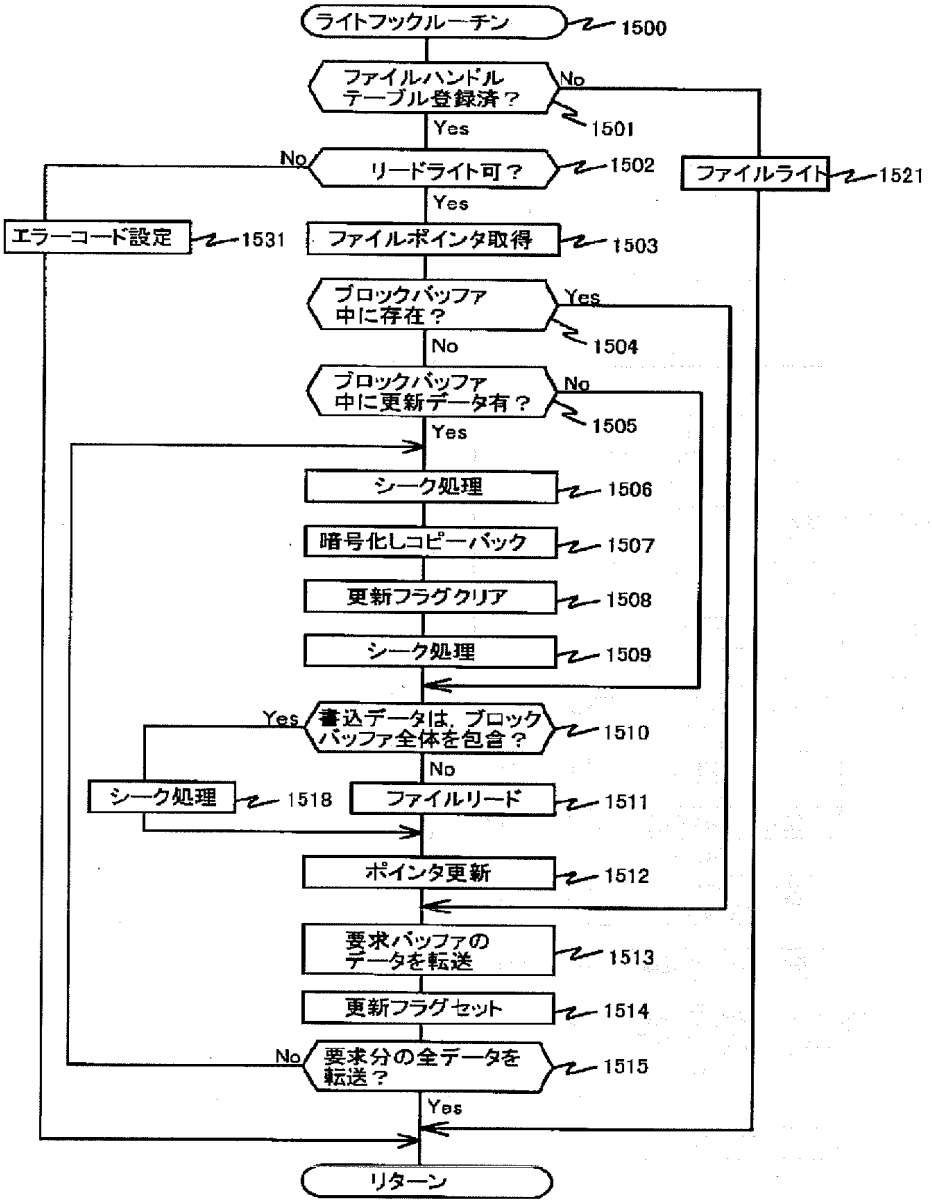
【図１４】


リードフックルーチン
1400
ファイルハンドル
テーブル登録済？
No
1401
Yes
リード可？
No
1402
Yes
ファイルリード
1421
エラーコード設定
1422
ファイルポインタ取得
1403
ブロックバッファ
中に存在？
Yes
1404
No
1405
ブロックバッファ
中に更新データ有？
No
Yes
1406
シーク処理
1407
暗号化しコピーバック
1408
更新フラグクリア
1409
シーク処理
ファイルリード
1410
ポインタ更新
1411
復号化処理実行
1412
要求バッファへ
データを転送
1413
平文ポインタ更新
1414
No
要求分の全データを
要求バッファへ転送？
1415
Yes
リターン

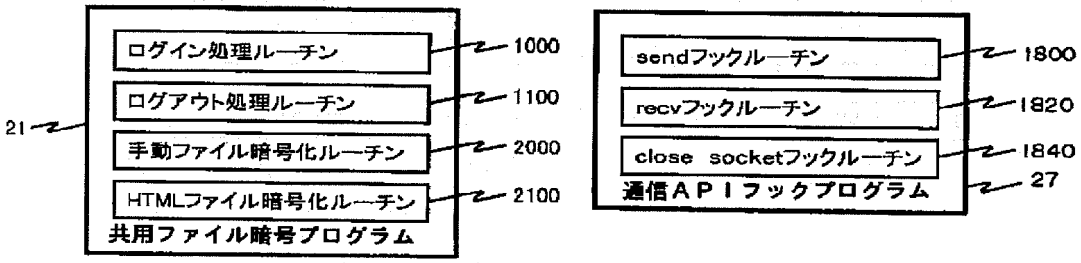
【図２７】


ログイン処理ルーチン
1000
ログアウト処理ルーチン
1100
21
手動ファイル暗号化ルーチン
2000
HTMLファイル暗号化ルーチン
2100
共用ファイル暗号プログラム
ファイルオープンフックルーチン
1200
ファイルクリエイトフックルーチン
1300
ファイルリードフックルーチン
1400
ファイルライトフックルーチン
1500
ファイルクローズフックルーチン
1600
22
ファイルリネームフックルーチン
1700
ファイルＩ／Ｄフックプログラム

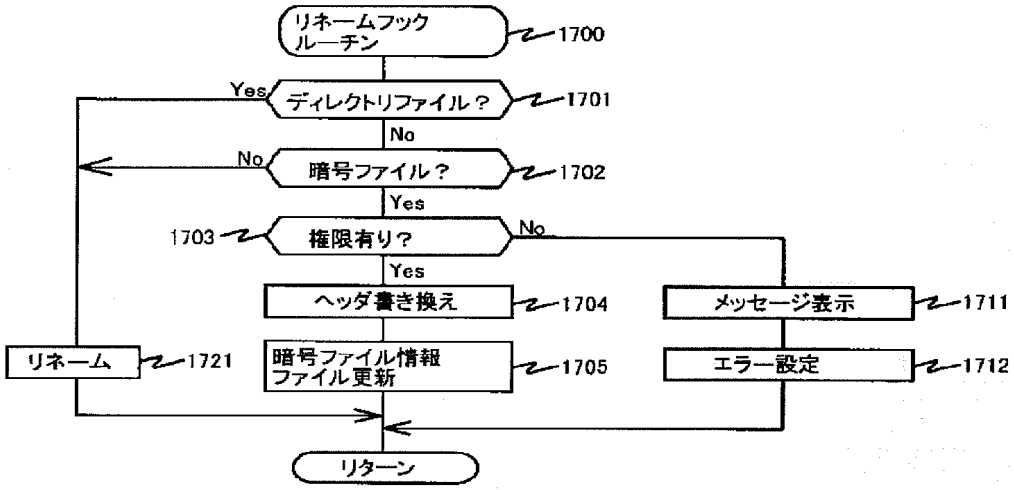
【図１５】


ライトフックルーチン
1500
ファイルハンドル テーブル登録済？
No
1501
Yes
No
リードライト可？
1502
ファイルライト
1521
Yes
エラーコード設定
1531
ファイルポインタ取得
1503
ブロックバッファ 中に存在？
Yes
1504
No
ブロックバッファ 中に更新データ有？
No
1505
Yes
シーク処理
1506
暗号化しコピーバック
1507
更新フラグクリア
1508
シーク処理
1509
Yes
書込データは，ブロック バッファ全体を包含？
1510
No
シーク処理
1518
ファイルリード
1511
ポインタ更新
1512
要求バッファの データを転送
1513
更新フラグセット
1514
No
要求分の全データを 転送？
1515
Yes
リターン

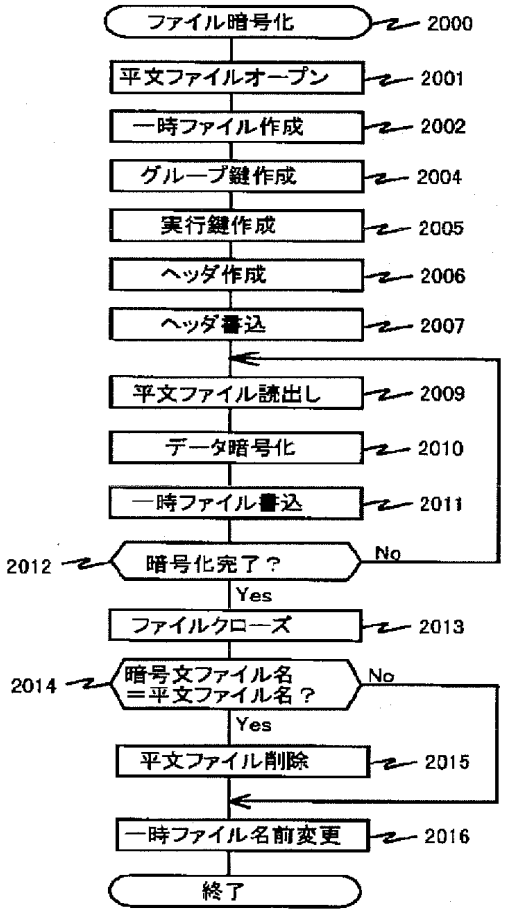
【図２９】


ログイン処理ルーチン
1000
ログアウト処理ルーチン
1100
手動ファイル暗号化ルーチン
2000
HTMLファイル暗号化ルーチン
2100
21
共用ファイル暗号プログラム
sendフックルーチン
1800
recvフックルーチン
1820
close socketフックルーチン
1840
通信ＡＰＩフックプログラム
27

【図１７】



【図２０】



【図２４】

secret.htm

```
<HTML>
<HEAD><TITLE>Sales information</TITLE></HEAD>
<BODY>
このページは，暗号化されています。
復号には，グループ暗号ソフトウェアと復号権限が必要です。
<!--GroupEncryption V1.0JKasdHfDsLWHewjkdU
ILAYHq3eu3'(UIyuak.f4n48hjBflse.a489basdJK
LSHDskdakSJDnds;3wKJDSDKosidja()IOsd2heHWQ
whewqhOQW+EH;o.ahue.hweo89seSJDLA+JD9uiJda
klu23eklau09hhjss;lqwen24 vkljdallana/\;ak
jD)DIj#L4h3nq-->
</BODY>
</HTML>
```

2401

```
<HTML>
<HEAD><TITLE>Sales information</TITLE></HEAD>
<BODY>
このページは，暗号化されています。
復号には，グループ暗号ソフトウェアと復号権限が必要です。
<!--
```

2402

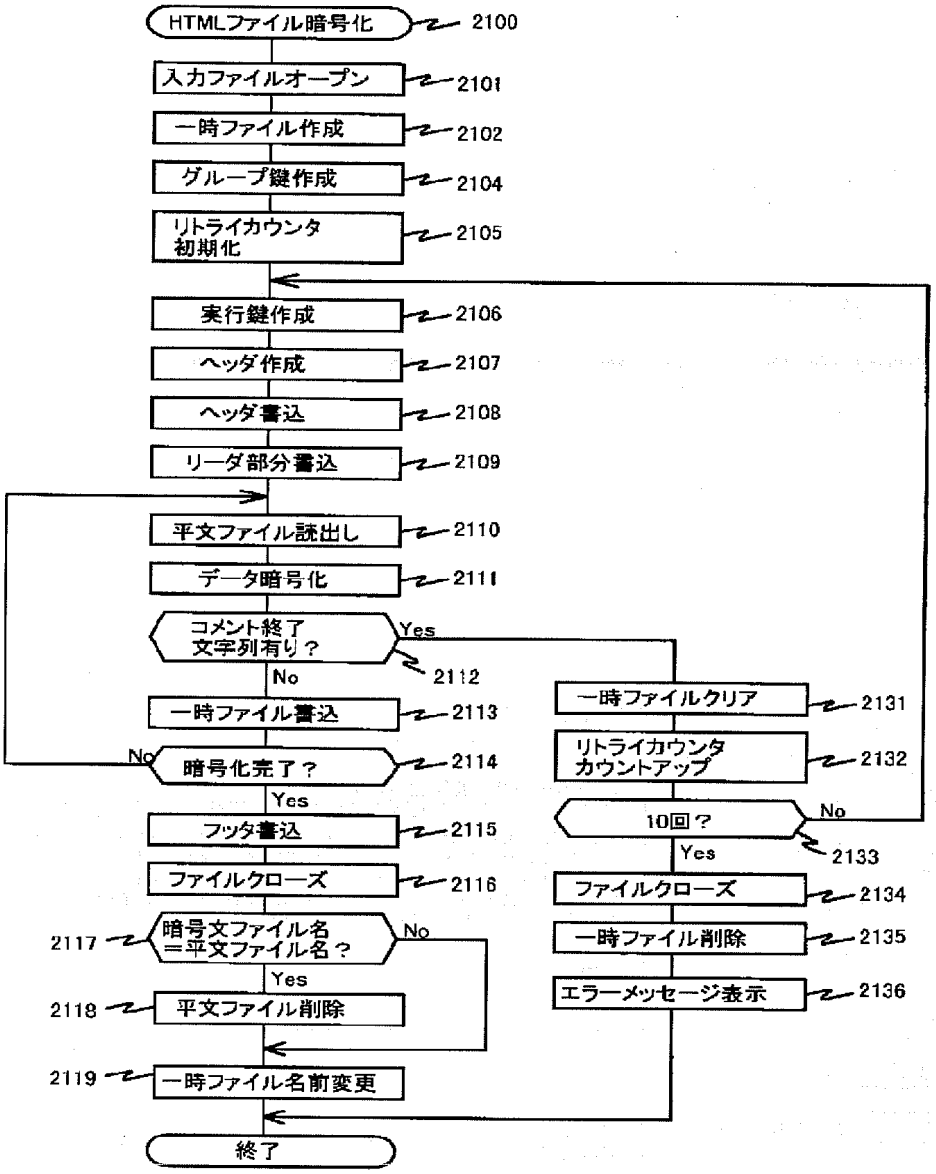
```
GroupEncryption V1.0JKasdHfDsLWHewjkdU
ILAYHq3eu3'(UIyuak.f4n48hjBflse.a489basdJK
LSHDskdakSJDnds;3wKJDSDKosidja()IOsd2heHWQ
whewqhOQW+EH;o.ahue.hweo89seSJDLA+JD9uiJda
klu23eklau09hhjss;lqwen24 vkljdallana/\;ak
jD)DIj#L4h3nq
```

2403

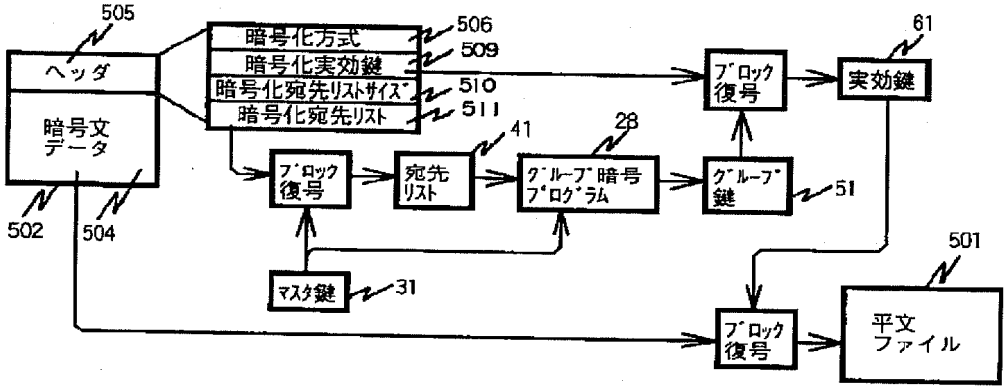
```
-->
</BODY>
</HTML>
```

2404

【図２１】


HTMLファイル暗号化
2100
入力ファイルオープン
2101
一時ファイル作成
2102
グループ鍵作成
2104
リトライカウンタ初期化
2105
実行鍵作成
2106
ヘッダ作成
2107
ヘッダ書込
2108
リーダ部分書込
2109
平文ファイル読出し
2110
データ暗号化
2111
コメント終了文字列有り？
Yes
2112
No
一時ファイル書込
2113
No
暗号化完了？
2114
Yes
フッタ書込
2115
ファイルクローズ
2116
2117
暗号文ファイル名＝平文ファイル名？
No
Yes
2118
平文ファイル削除
2119
一時ファイル名前変更
終了
一時ファイルクリア
2131
リトライカウンタカウントアップ
2132
10回？
No
2133
Yes
ファイルクローズ
2134
一時ファイル削除
2135
エラーメッセージ表示
2136

【図３１】


505
ヘッダ
暗号文データ
502
504
暗号化方式
506
暗号化実効鍵
509
暗号化宛先リストサイズ
510
暗号化宛先リスト
511
ﾌﾞﾛｯｸ復号
41
宛先リスト
28
ｸﾞﾙｰﾌﾟ暗号ﾌﾟﾛｸﾞﾗﾑ
ﾌﾞﾛｯｸ復号
61
実効鍵
ｸﾞﾙｰﾌﾟ鍵
51
ﾏｽﾀ鍵
31
501
ﾌﾞﾛｯｸ復号
平文ファイル

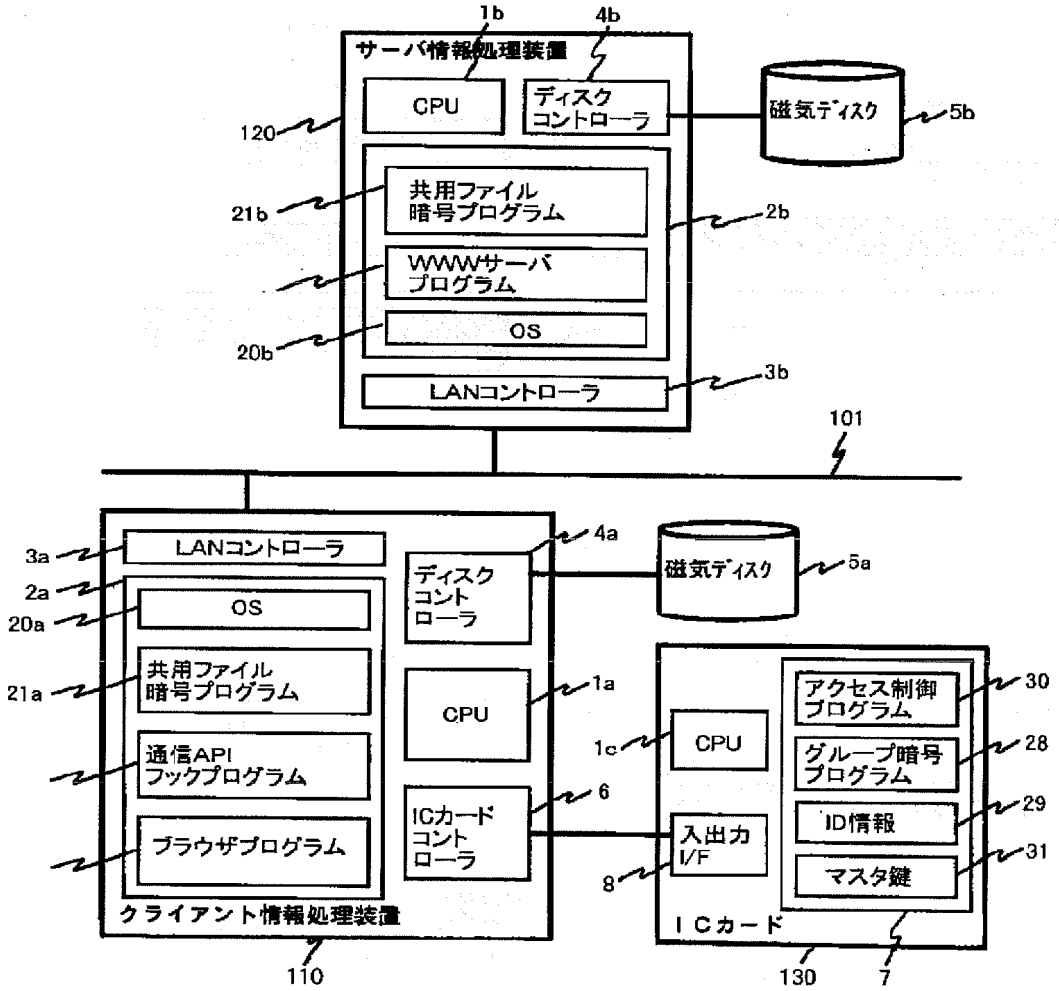
【図２６】

secret.htm

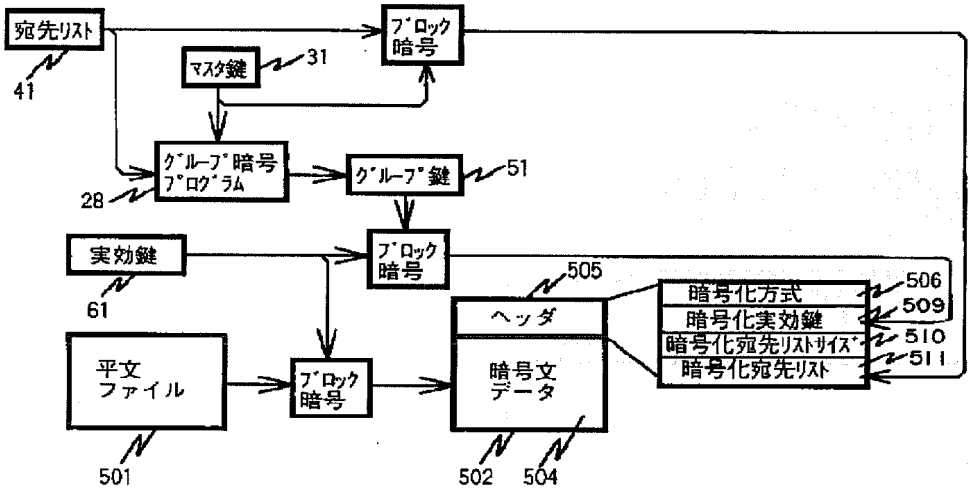
```
<HTML>
<HEAD><TITLE>Sales information</TITLE></HEAD>
<BODY>
このページは，暗号化されています。
復号には，グループ暗号ソフトウェアと復号権限が必要です。

<!--GroupEncryption V1.04A4B61736448664473
4C574865776A6B6455494C4159487133657533270D
0A2855497975616B2E66346E3438686A42666C7365
2E6134383962617364 4A4B4C534844736B64616B53
4A446E64733B33774B4A440D0A53444B6F7369646A
612829494F73643268654857517768657771684F51
572B45483B6F2E616875652E6877656F383973650D
0A534A444C412B4A443975694A64616B6C75323365
6B6C6175303968686A73733B6C7177656E32342076
6B6C6A64616C6C616E612F5C3B0D0A616B6A442944
496A234C3468336E71-->

</BODY>
</HTML>
```

2601

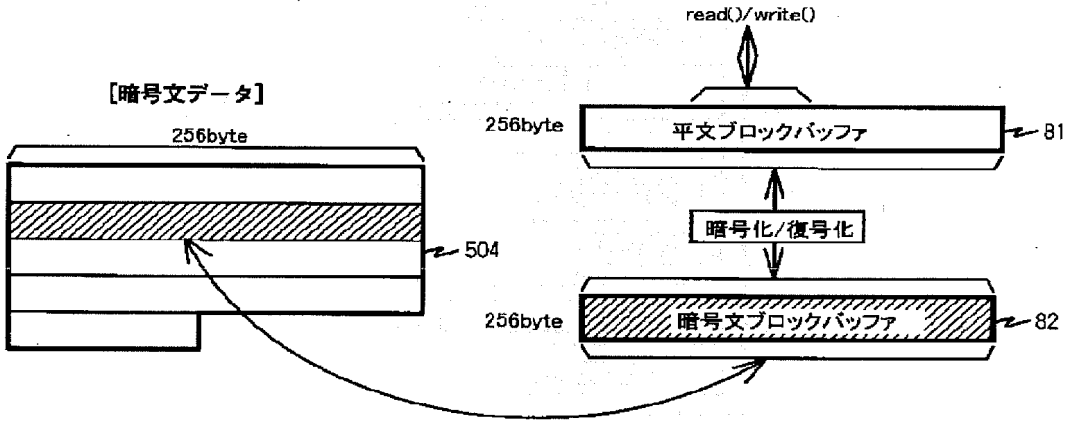
【図２８】



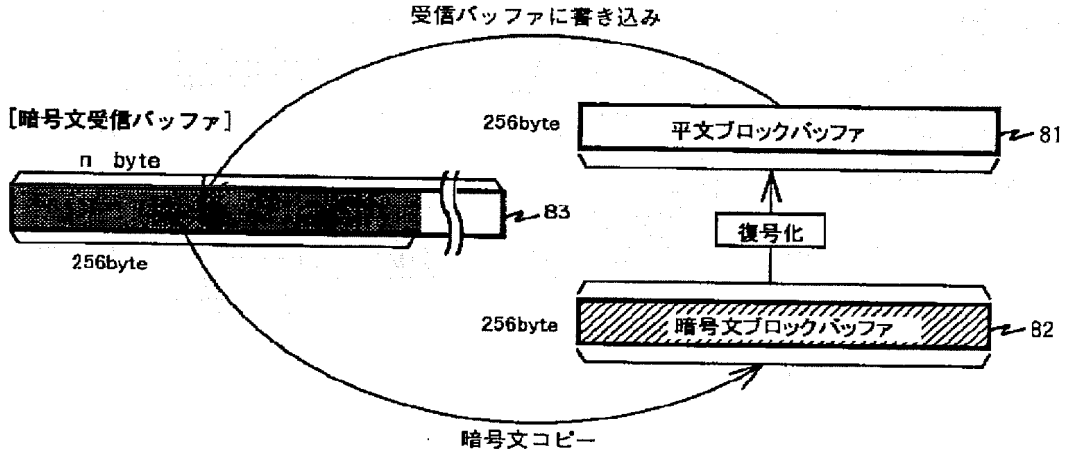
【図３０】


宛先リスト
41
ﾌﾞﾛｯｸ暗号
ﾏｽﾀ鍵
31
ｸﾞﾙｰﾌﾟ暗号ﾌﾟﾛｸﾞﾗﾑ
28
ｸﾞﾙｰﾌﾟ鍵
51
実効鍵
61
ﾌﾞﾛｯｸ暗号
505
ヘッダ
暗号化方式
506
暗号化実効鍵
509
暗号化宛先ﾘｽﾄｻｲｽﾞ
510
暗号化宛先リスト
511
平文ファイル
501
ﾌﾞﾛｯｸ暗号
暗号文データ
502
504

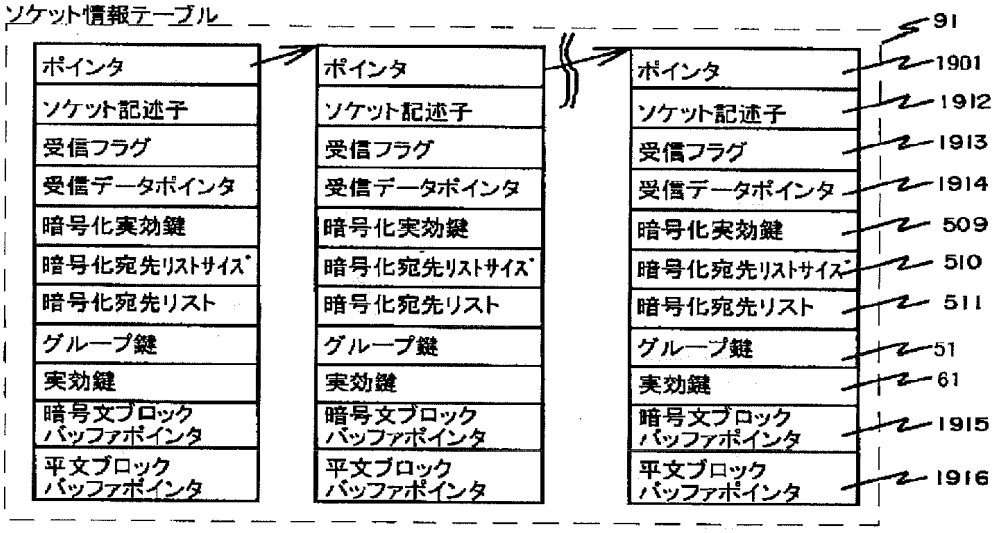
【図３２】


read()/write()
[暗号文データ]
256byte
504
256byte
平文ブロックバッファ
81
暗号化/復号化
256byte
暗号文ブロックバッファ
82

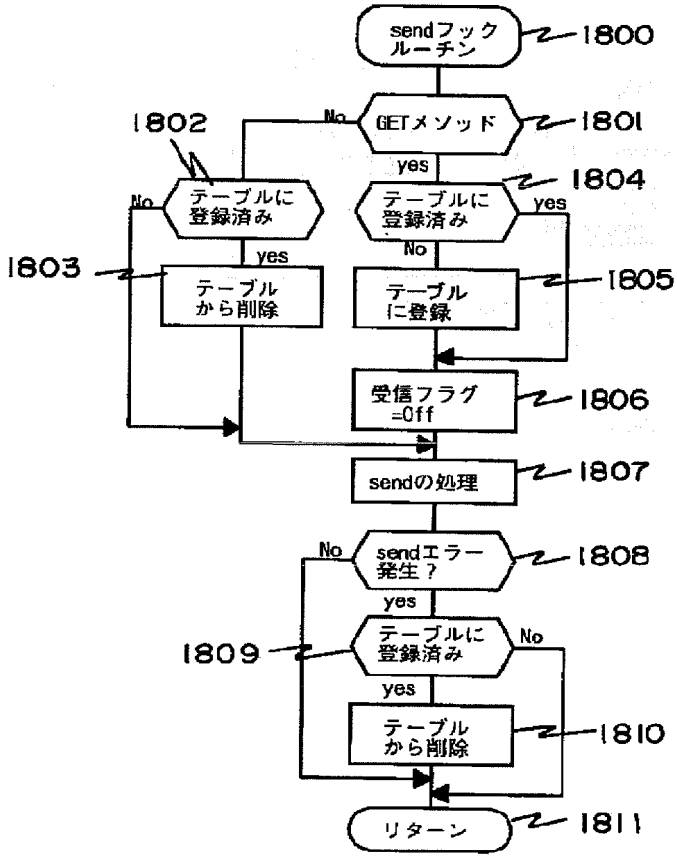
【図３３】


受信バッファに書き込み
[暗号文受信バッファ]
n byte
83
256byte
256byte
平文ブロックバッファ
81
復号化
256byte
暗号文ブロックバッファ
82
暗号文コピー

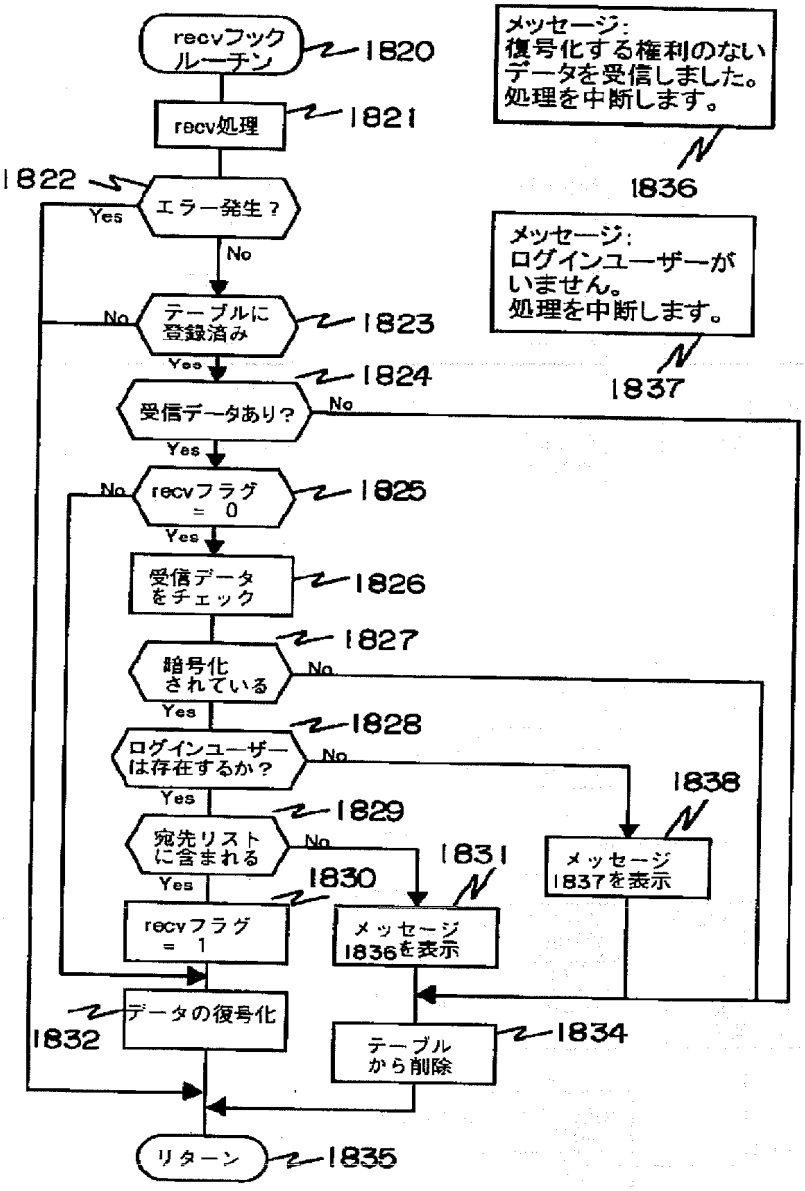
【図３４】


ソケット情報テーブル
91
ポインタ
ソケット記述子
受信フラグ
受信データポインタ
暗号化実効鍵
暗号化宛先リストサイズ
暗号化宛先リスト
グループ鍵
実効鍵
暗号文ブロック
バッファポインタ
平文ブロック
バッファポインタ
ポインタ
ソケット記述子
受信フラグ
受信データポインタ
暗号化実効鍵
暗号化宛先リストサイズ
暗号化宛先リスト
グループ鍵
実効鍵
暗号文ブロック
バッファポインタ
平文ブロック
バッファポインタ
ポインタ
1901
ソケット記述子
1912
受信フラグ
1913
受信データポインタ
1914
暗号化実効鍵
509
暗号化宛先リストサイズ
510
暗号化宛先リスト
511
グループ鍵
51
実効鍵
61
暗号文ブロック
バッファポインタ
1915
平文ブロック
バッファポインタ
1916

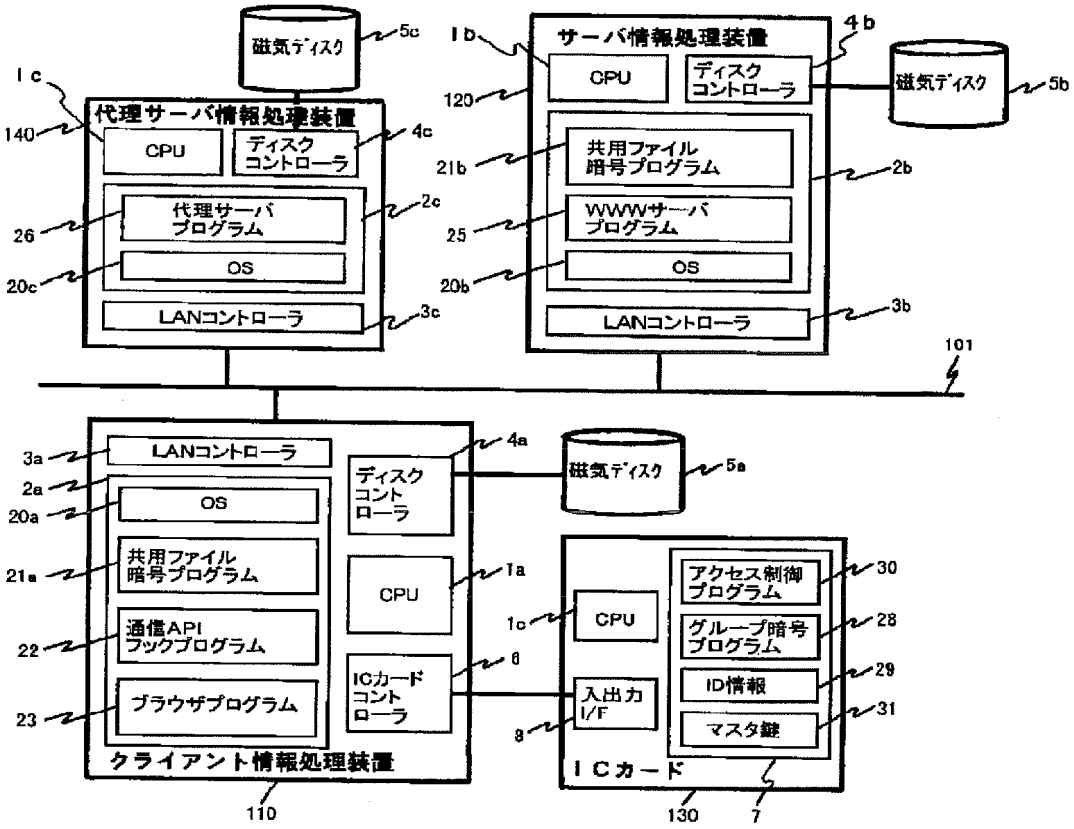
【図３５】


sendフック
ルーチン
1800
GETメソッド
1801
No
yes
1802
テーブルに
登録済み
No
yes
1803
テーブル
から削除
1804
テーブルに
登録済み
yes
No
テーブル
に登録
1805
受信フラグ
=Off
1806
sendの処理
1807
No
sendエラー
発生？
1808
yes
1809
テーブルに
登録済み
No
yes
テーブル
から削除
1810
リターン
1811

【図３６】


recvフック
ルーチン
1820
recv処理
1821
1822
エラー発生？
Yes
No
テーブルに
登録済み
1823
No
Yes
1824
受信データあり？
No
Yes
recvフラグ
＝ 0
1825
No
Yes
受信データ
をチェック
1826
1827
暗号化
されている
No
Yes
1828
ログインユーザー
は存在するか？
No
Yes
1829
宛先リスト
に含まれる
No
Yes
1830
recvフラグ
＝ 1
1831
メッセージ
1836を表示
1838
メッセージ
1837を表示
1832
データの復号化
テーブル
から削除
1834
リターン
1835
メッセージ:
復号化する権利のない
データを受信しました。
処理を中断します。
1836
メッセージ:
ログインユーザーが
いません。
処理を中断します。
1837

【図３８】



フロントページの続き

(72) 発明者　梅木　久志
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内
(72) 発明者　大津　豊
神奈川県横浜市戸塚区戸塚町5030番地　株式会社日立製作所ソフトウェア開発本部内
(72) 発明者　森藤　元
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内
(72) 発明者　清水　麻由子
神奈川県横浜市戸塚区戸塚町5030番地　株式会社日立製作所ソフトウェア開発本部内